PIVOTAL AND THE THREAT OF OPEN SOURCE CRM: A STRATEGIC ASSESSMENT

by

Matthew Brownstein
Diego Lin

PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF BUSINESS ADMINISTRATION

In the
Faculty
of
Business Administration

Management of Technology MBA

SIMON FRASER UNIVERSITY

Fall 2005

APPROVAL

Name: **Matthew Brownstein**
Diego Lin

Degree: **Master of Business Administration**

Title of Project: **PIVOTAL AND THE THREAT FROM OPEN SOURCE: A STRATEGIC ASSESSMENT**

Supervisory Committee:

____

Dr. Aidan Vining
Senior Supervisor
Professor, Faculty of Business Administration

____

Dr. Michael Brydon
Second Reader
Assistant Professor, Faculty of Business of Administration

Date Approved: December 5, 2005

SIMON FRASER UNIVERSITY

Ethics Approval

The authors, whose names appears on the title page of this work, have obtained human research ethics approval from the Simon Fraser University Office of Research Ethics for the research described in this work, or have conducted the research as co-investigators of a project, or members of a course, approved by the Ethics Office.

A copy of the human research ethics approval letter has been filed at the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for ethics approval and letter of approval is filed with the Office of Research Ethics. Inquiries may be directed to that Office.

W. A. C. Bennett Library
Simon Fraser University
Burnaby, BC, Canada

ABSTRACT

The open source idea is gaining currency. In areas such as scripting languages, web server platforms, and database servers, open source software has managed to compete with and surpass commercial incumbents. Could the CRM industry be next? To answer this question, we introduce four open source case studies: Linux, Apache, MySQL, and PHP. Identifying commonalities between them, we suggest a framework for explaining the success of open source projects. Based on this framework, we evaluate the likelihood of open source CRM succeeding and pinpoint future developments that would trigger its widespread adoption. We conclude that open source CRM is likely to succeed and that its success will negatively impact commercial CRM vendors. In light of these conclusions, we propose recommendations that Pivotal Corporation – a commercial CRM incumbent – can pursue in order to better position itself in the industry's evolving competitive landscape.

Keywords: *Open Source Software; Customer Relationship Management; CRM Industry; Disruptive Technology; Pivotal Corporation; SugarCRM; Provision of Public Goods*

ACKNOWLEDGEMENTS

We would like to thank the teaching and administrative staff of Simon Fraser University's Management of Technology MBA program for their help and guidance. Our supervisory committee, consisting of Dr. Aidan Vining and Dr. Michael Brydon, provided helpful and timely feedback, for which we are grateful.

We would also like to extend our gratitude to Pivotal Corporation and to our project sponsor, Senior Director of Product Management, Scott Munro, for their kind support and hospitality.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

CRM	Customer Relationship Management
SMB	Small-to-medium size businesses
GNU	Free operating system; recursive acronym for “GNU’s not Unix”
GPL	GNU General Public License
FSF	Free Software Foundation
LAMP	Linux-Apache-MySQL and either PHP, Perl, or Python
TCO	Total Cost of Ownership

1 INTRODUCTION

The purpose of this paper is to assess open source CRM's likely impact on the CRM industry. Does Pivotal Corporation, as a commercial CRM software company, need to worry about open source CRM? On the surface, the answer is no. Despite impressive growth in downloads and users, open source CRM has made little market impact. Other open source projects, however, have successfully overtaken commercial firms in their respective industries. We examine the cases of the Linux operating system, Apache web server, MySQL database server, and PHP scripting language. Are such cases analogous to open source CRM? Or is there perhaps something that sets CRM apart from the industries affected in these four examples?

Our approach to answering these questions leads us down two complementary paths. First, we investigate the CRM industry's past and present for clues that could point towards its future. If a shift towards open source is to occur, there would need to be sufficient impetus for change. Understanding recent trends and developments will shed light on where the industry is likely headed. Second, we pinpoint future events that would lead to widespread adoption of open source CRM software. Assessing the likelihood of these events will enable us to gauge the seriousness of the threat. Based on the conclusions reached, we evaluate several strategic options Pivotal could reasonably pursue in order to better position itself in the CRM industry's evolving competitive landscape.

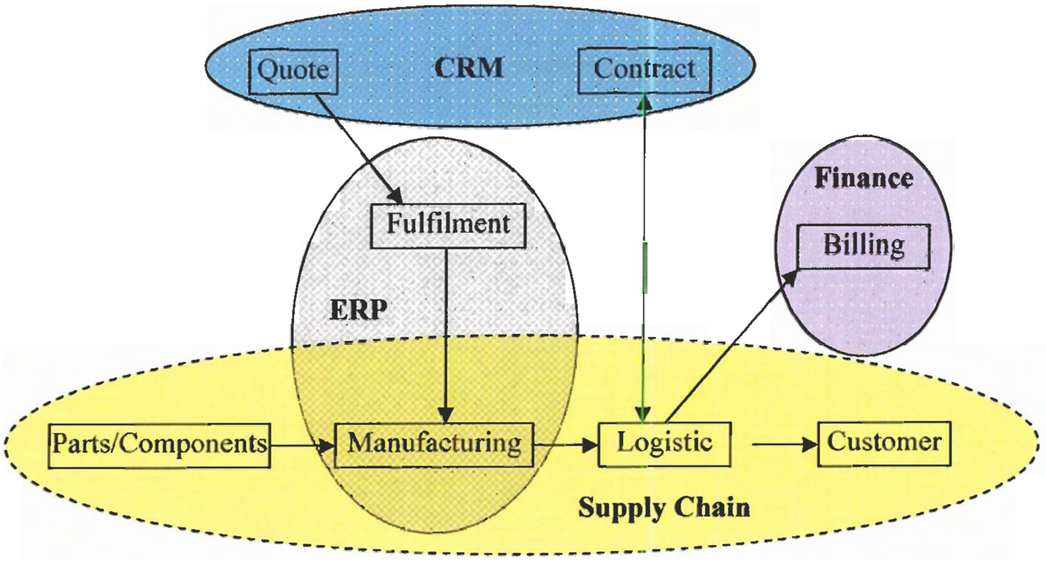
2 PIVOTAL'S CURRENT SITUATION

Three central topics form the basis of our analyses: the CRM industry, Pivotal's place within it, and the emergence of open source CRM. This section serves to introduce all three.

2.1 The CRM Industry on the Rebound

As its name implies, Customer Relationship Management (CRM) software helps companies manage and improve customer interactions. In the context of an organisation's supply chain, CRM centers upon the managing of quotes and contracts (see *Figure 2.1*). Though sometimes grouped together with ERP software, the two are generally considered distinct applications – each focusing on slightly different pieces of an organisation's Information Technology puzzle.

Figure 2.1 CRM's Place within a Firm's Technology Architecture[1]



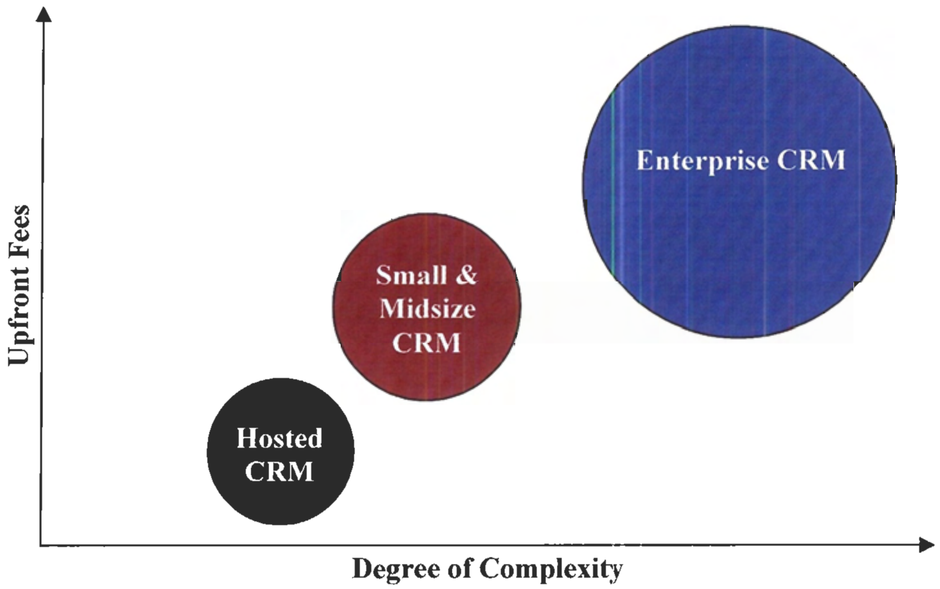
The CRM industry is characterised by two prominent strategic groups.[2] The first group consists primarily of the big three CRM vendors: SAP, Oracle, and Siebel (recently acquired by Oracle). Their sights are set squarely on what we are calling the "Fortune 500" segment of the market – that is, large, successful organisations. Product offerings from the big three vendors involve massive implementation projects. They are characterised by a high degree of complexity and large upfront fees (see *Figure 2.2*). The second group is far more fragmented. It consists of myriad small CRM vendors targeting small-to-medium size businesses (SMB). Pivotal, Onyx, and SalesLogix are the market leaders in this group (Close, 2003). A third group has recently emerged. Led by Salesforce.com, it consists of "hosted" CRM vendors. Instead of implementing CRM software locally on the customer's end, hosted vendors offer "on demand" CRM use via

[1] Based on a conversation with Pivotal's Senior Product Marketing Manager, Dana Crane, September 28, 2005.

[2] A strategic group is "a cluster of firms within an industry that follow strategies common to the group but distinctive from firms in other groups." (Oster, 1999)

the Internet. Their focus is also geared towards the SMB segment, with particular emphasis on the "non-user" (i.e. SMBs that have yet to adopt any CRM product).

Figure 2.2 Strategic Groups in the CRM Industry[3]



The industry as a whole is highly fragmented. Even for the big three vendors, total market share ranges from only 6.7% to 11.9% (Wardley & Bulmstein, 2004). For providers in the second and third groups, market share is, of course, even lower.

Table 2.1 CRM Market Share and Growth Comparison

Company	2003 Market Share	Growth (2002 to 2003)
Siebel Systems	11.9%	-26.30%
SAP	6.7%	+15.60%
Oracle	7.5%	-3.7%

[3] The size of the circles corresponds to each group's relative market share.

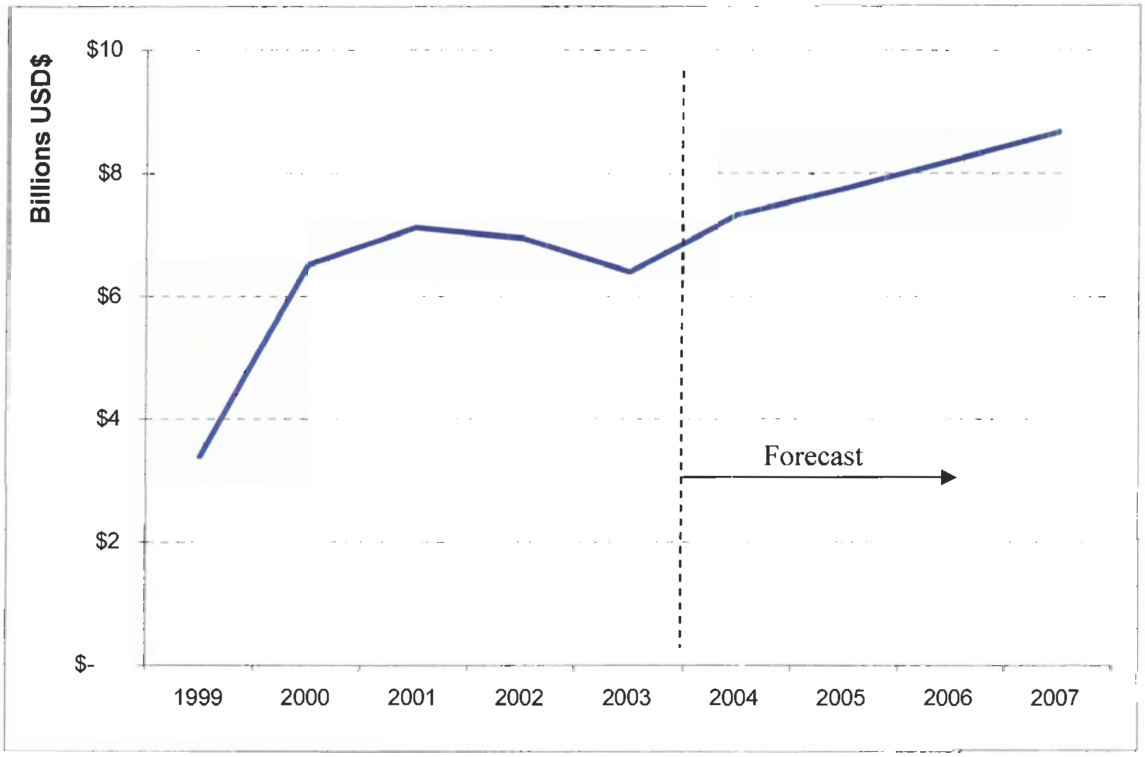
Company	2003 Market Share	Growth (2002 to 2003)
Microsoft	1.4%	+90.12%
SalesLogix	1.1%	+18.5%
Pivotal	0.4%	-38.77%
Onyx	0.4%	-24.06%
Salesforce.com	0.3%	+84.92%

Source: Based on Wardley & Bulmstein, 2004

During the 1990s, the CRM industry was characterised by hype; double-digit growth was the norm. Like most technology-related industries, however, the CRM industry was hit hard by the "tech bubble burst" of 2000-2002. As financially constrained customers sought to save money, the CRM industry bore the brunt of their budgetary cutbacks. The second and third groups were impacted most, since SMB customers are especially vulnerable to market fluctuations. CRM vendors that had jumped on the CRM bandwagon during the 90s without sound business models could no longer compete in the new environment. This gave rise to a winnowing out process in which a number of smaller CRM vendors went bankrupt.[4] Since then, industry growth has ranged from slow to stagnant. The last couple of years have shown signs of an industry-wide upswing.

[4] Bruce Kenny, Pivotal's former Senior Vice President of Products. SFU Executive Speaker, March 4, 2005.

Figure 2.3 CRM Industry Growth[5]

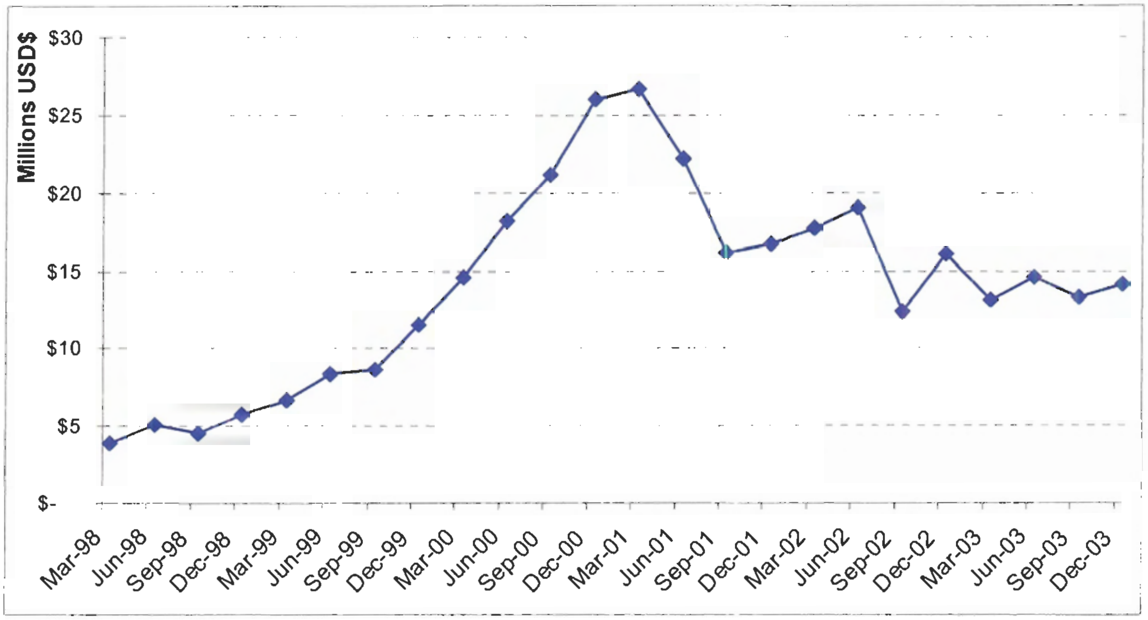


2.2 Pivotal's Comfortable Niche Position

Pivotal holds a strong niche position in the CRM industry. In competing over the lucrative "Fortune 500" segment of the market, the big CRM vendors price themselves out of range of most SMBs (Band et al., 2005). Pivotal and a host of smaller CRM vendors capitalise by filling the void. Additionally, there are substantial costs associated with switching between CRM providers, which serve to "lock-in" customers. Taken together, these two factors make Pivotal's current position as a reputable CRM vendor with an established customer base an enviable one.

[5] Based on data and forecasts provided in Wardley & Bulmstein (2004).

Figure 2.4 Pivotal's Revenue Growth[6]



The industry-wide downturn of 2000-2002 is mirrored in Pivotal's sales data. Total revenues fell by 116% from their peak in March 2001 to their nadir in September 2002. Signs of an industry turnaround are present in Pivotal's sales figures as well. Revenues stabilised in 2003 and increased steadily in 2004 and 2005. [7] On the whole, Pivotal seems poised to capitalise on the industry's rebound.

2.3 Troubles on the Horizon?

Multiple open source CRM projects have surfaced worldwide within the past two years.[8] Each is in need of a critical mass of users before it can truly emerge as an alternative to commercial products. SugarCRM, founded in April 2004, is currently the clear frontrunner. Interestingly, though SugarCRM's software application is open source,

[6] See *Appendix B* for detailed revenue data.
[7] Pivotal was acquired by CDC in 2003. Financial data post-acquisition is grouped together with CDC's other CRM/ERP interests, and therefore is not comparable to financial data pre-acquisition.
[8] Spanish-based Hipergate (http://www.hipergate.org), Indian-based vtiger (http://www.vtiger.com), and US-based Compiere (http://www.compiere.com), to name a few.

SugarCRM is a profit-driven firm. The company's revenue model consists in selling advanced add-on products, customisation services, and customer support. SugarCRM secured over $2 million in venture capital seed funding, which enabled the company to quickly establish itself as the prominent name in open source CRM. Between 2004 and 2005, its customer base skyrocketed from 50 customers to over 2,000. Most of these customers are very small organisations, however – even by SMB standards. As a result, SugarCRM and the other open source CRM projects have to date made little market impact.

From Pivotal's vantage point, it is difficult to take open source CRM very seriously. SugarCRM's offering falls significantly short of Pivotal's in terms of functionality, usability, and sophistication. Pivotal acknowledges that open source CRM barely even registers on its competitive radar.[9] Why should it? Given its industry position, it makes more sense for Pivotal to focus on building market share. The other niche CRM providers competing in the SMB arena, such as Onyx and SalesLogix, are of primary concern. Open source CRM, on the other hand, lies on the periphery. In the countless Gartner CRM industry reports, there is scarcely even a brief mention of open source CRM.[10]

All of which begs the question: Why have we opted to make the "threat" of open source CRM the centerpiece of our analysis? Despite its apparent insignificance, we argue there is good reason for taking open source CRM seriously. Open source software has demonstrated an ability to compete with and, indeed, surpass incumbents in such

[9] Conversation with Pivotal's Senior Director of Product Management, Scott Munro, Sept. 21, 2005.
[10] Gartner's 50-page 2005 CRM Vendor Guide, for example, does not contain a single reference to open source or to any open source CRM vendors.

areas as scripting languages, web server platforms, and database servers. Could the CRM industry be next? To answer this question, we begin by providing some background on the open source idea.

3 THE OPEN SOURCE IDEA

The emergence of the Linux operating system in the 1990s popularised the notion of open source. It became almost fashionable. To many, it represented a long sought after alternative – an alternative to the dominant software giants, to Microsoft, perhaps even to capitalism itself. In the midst of such popularity, it is easy to lose sight of what the term actually means.

3.1 The Meaning of "Open Source"

"Open source" is used in two distinct, yet related contexts: (1) as an ideology and (2) as a design methodology. In the first instance, the meaning is that a software program's underlying source code *should* be made freely available to users; in the second, that the process of designing the software *should* involve as many people as possible – experts and non-experts alike – in the belief that the small contributions of many will solve problems more effectively than the full-time dedication of a few.

The commonality in both designations is the importance placed on shared knowledge. Since users encounter similar problems, and devise similar solutions, sharing this knowledge prevents others from having to "re-invent the wheel" (Hippel, 2005). The sharing of private knowledge, it is held, increases overall creativity and innovation, thereby benefiting all.

3.2 The History of Open Source

The open source idea emerged in the early 1960s with the implementation of time-sharing systems at research labs across the United States. In a time-sharing system, a single computer is used to provide interactive general-purpose computing to multiple users by sharing processor time. Their development gave rise to the first *communities* of computer users. The potential benefits associated with sharing knowledge with other members of the community soon became apparent. Robert Fano, an architect of MIT's first time-sharing system noted that "more than half of the current system commands in the compatible Time-Sharing System at MIT were developed by system users rather than by the system programmers responsible for the development and maintenance of the system" (Fano, 1967). Sharing small innovations and modifications amongst a community of like-minded users stretched the system's utility beyond what the system designers could have provided on their own.

Individuals investing the time and effort necessary to share their work with others did not receive any monetary compensation for doing so. From an economics standpoint, this is difficult to explain. One would expect instead a tendency towards free-riding. A simple cost-benefit assessment might lead community members to ask themselves: Why should I contribute when I can simply wait for others to do so while I reap the benefits? With enough members asking themselves the same question, community contributions would eventually dwindle to nothing. Yet Fano remarked that "[t]he opposite phenomenon seems to be occurring… many people do indeed invest the additional effort required to make their work usable by others" (Fano, 1967). Similar experiences were recounted at other research laboratories and academic institutions. This earliest coalescing of interests between open source philosophy and the budding computer revolution is important. It shaped and coloured the environment in which the open source idea would later re-emerge in the 1980s and 90s.

The 1970s saw the birth of the modern concept of proprietary software. Previously, hardware and software were bundled together. When the computer-equipment industry opted to "unbundle" them, the trend shifted towards developing software for industry-standard platforms. The realisation that the software could generate revenue soon followed, and independent software firms began to surface (Tuomi, 2005).

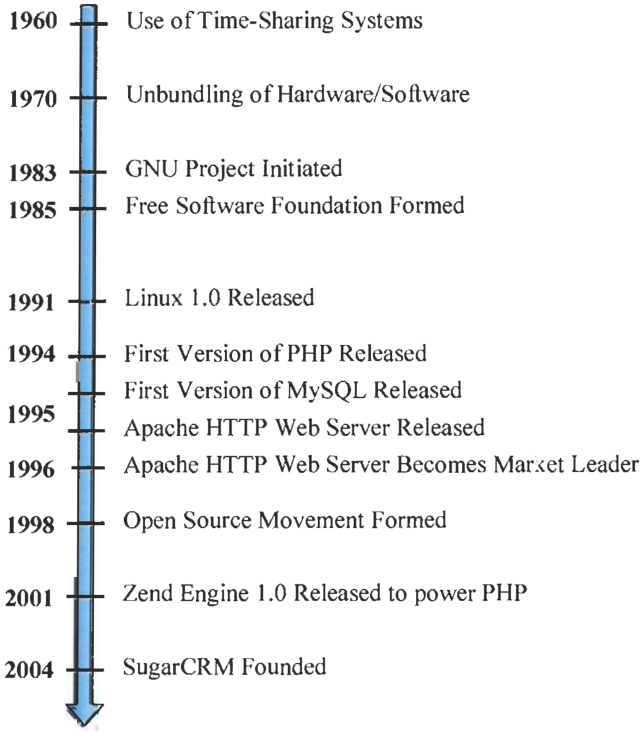
Given the value of this more-or-less intangible good, software companies placed heavy emphasis on protecting their intellectual property. A closed lid was kept over software's most tangible aspect: its underlying series of 0s and 1s. Doing so, however, often frustrated software developers. Restricted access to source code made it hard to interface amongst different proprietary applications. Tuomi (2005) pointed out the difficulty of "integrating peripheral equipment, such as printers, into developed systems" during this period (Tuomi, 2005). Ironically, it is this very frustration with proprietary software that gave legs to the open source idea in its early stages.

3.3 "Free" Software

Richard Stallman founded the Free Software Foundation in 1985 in response to such frustrations. His pioneering contribution was adopting copyright law to promote free access to source code. Instead of using copyright law to restrict the right to make and redistribute copies of a particular work, he used it to ensure that copies and anything derived from them would continue to be used, modified, and shared with others. This novel twist to copyright licensing would later be termed "copyleft" licensing. Without a copyleft license, software code released to the public domain can be modified and redistributed as proprietary software. Stallman's GNU General Public License (GPL) prevents this activity by ensuring that derived works must continue to be made available to the public.

Stallman's Free Software Movement was imbued with a sense of romantic idealism. Many of its proponents held that it was inherently wrong and immoral to keep software code proprietary (Bonaccorsi, 2004). This anti-establishment, revolutionary ethos did not sit well with some members. In 1998, several prominent members left the Free Software Movement to form their own group.

Figure 3.1 Timeline of Open Source Software's Development


1960 Use of Time-Sharing Systems
1970 Unbundling of Hardware/Software
1983 GNU Project Initiated
1985 Free Software Foundation Formed
1991 Linux 1.0 Released
1994 First Version of PHP Released
First Version of MySQL Released
1995
Apache HTTP Web Server Released
1996 Apache HTTP Web Server Becomes Market Leader
1998 Open Source Movement Formed
2001 Zend Engine 1.0 Released to power PHP
2004 SugarCRM Founded

3.4 The Open Source Movement

Founded by John Hall, Larry Augustin, Eric Raymond, Bruce Perens, and others, the Open Source Movement advocated open source on pragmatic rather than on philosophical grounds. Software companies should employ the open source idea, it was argued, not because it would be morally wrong not to, but because of the practical advantages of open source software. Three primary claims were put forth for why supporting open source software makes good business sense, both for customers and for suppliers: (1) it lowers the costs of development and of purchase; (2) it results in better quality software; and (3) it allows for greater customisation.

The first claim – that open source software is less costly – is based on the fact that, unlike proprietary software, there are no upfront license fees to pay. Whether or not open source software is truly more cost effective in the long run is open to debate. Disputing the claim that Linux offers cost savings over Windows, for example, a Microsoft funded study showed that the total cost of ownership of Linux exceeds that of Windows (IDC, 2002). Subsequent studies dispute this finding, however.[11] The claim that open source results in better quality software is interesting. Since there is greater transparency throughout the development process, there is greater opportunity for peer review. More eyes scrutinising the code should, theoretically at least, result in higher quality software. Despite the apparent logic of this claim, there is a lack of evidence to either prove or disprove it. The claim is more anecdotal than empirical. The third claim appears to have merit. Access to source code makes it far easier to modify an application

[11] A 2004 study by Melbourne-based research firm, Cybersource, claims 36% savings for Linux over Windows for a 250-seat company (Cybersource, 2004). A 2004 study by Soeren Research, based on interviews with 50 organisations, corroborates this finding.

to suit particular needs. It is also helpful for interfacing a new application with an existing system. Proprietary software companies are sometimes compelled to open up limited parts of their source code to customers for this very reason.

The importance of the Open Source Movement lies principally in its bringing business and ideological interests together. Rational arguments were adduced for why the open source approach is better from a business perspective. One could of course question the soundness of the arguments, but the debate was at least being framed in terms the business community could understand. From there, it was not a stretch for profit-driven companies to begin thinking strategically about open source and about how it could be used to their advantage.

3.5 A Movement in Transition

The open source idea is evolving. While the Open Source Movement continues to identify with its ideological past, it has also paved the way for the commercialisation of open source. The distinction between open source and proprietary is becoming less and less clear. New business models are emerging that, in varying degrees, combine elements of both. Still timid, unsure, these hybrid models are exploring new terrain. On the one hand, proprietary software companies are toying with integrating aspects of the open source philosophy into their own strategies. For some, this can mean changing their internal approach to software development. For others, it can mean selectively releasing proprietary software to the open source community.[12] On the other hand, companies already involved with open source are exploring ways of leveraging their reputation and

[12] *Section 5* discusses the motivations behind the calculated use of open source by profit-driven firms.

expertise for commercial gain. Given that the source code is free, complementors, such as accreditation, custom services, and training are taking on heightened significance.

3.6 Challenging Traditional Software Models

While questions about where the movement is ultimately headed remain unanswered, one thing is certain: open source software is more than just the passing fad it was once dismissed as being. The movement has outlived the predictions of critics claiming that it would not last. Moreover, its convergence with profit-driven motives has made it more commercially important than ever. The open source idea poses a challenge to traditional ways of thinking about software and software organisations. Two fundamental differences warrant further consideration: its flexible revenue model and its open development process.

3.6.1 Flexible Revenue Model

A study by Rick Sherlund, technology sector analyst for Goldman Sachs, indicates that software companies allocate, on average, 76% of software licensing revenues towards their sales and marketing functions (cited in Augustine, 2005). In other words, the hefty license fees, prohibitive to large segments of customers, are spent convincing customers to purchase the software in the first place.

The open source model takes a different approach. It shifts the focus away from selling software, placing it instead on selling the ability to modify software to address specific customer needs. The code is openly available. Customers are free to download or install it and, having done so, to decide what modifications are needed. Without having to bankroll large sales and marketing departments, there is greater pricing flexibility.

Open source can therefore eliminate – or, at least, minimise – both the large license fees and the cumbersome sales departments. Without the licence fees, a significant, previously unattainable segment of the market opens up. Without the large sales departments, and the bureaucracy that accompanies them, resources can be allocated towards core activities. The end result is a more efficient, less unfocused software organisation.

3.6.2 Open Development Process

In addition to opening the door to new, potentially more efficient revenue models, open source challenges traditional conceptions about software development. Instead of an organisation's software development and debugging processes being the exclusive domain of a few high-paid specialists, open source philosophy holds that "given enough eyes, all bugs become shallow" (Raymond, 1999). The belief is that if enough people see an application's underlying source code – experts and non-experts alike – a better program will emerge than a few experts could divine on their own. The variegated experiences and backgrounds of many create synergies that make the whole greater than the parts.

Another aspect of open source's development methodology is its emphasis on user involvement. The line between developers and users is blurred; users become developers, funnelling their ideas and modifications back into the larger community. The benefits of such a development approach accrue particularly in markets requiring a large degree of customisation – provided the modifications are useful to others. Effectively, it enables companies to outsource part of their research and development costs to users.

From the perspective of users, this arrangement makes sense. It ensures their custom modifications do not become "orphaned" – that is, that they do not get left behind in future upgrades and releases.

These two elements – its flexible revenue model and open development process – represent open source software's fundamental challenge to commercial software companies. The objective here is not to polemicize for or against open source software. It is, rather, to point out that the open source model is both different and viable. It presents an alternative to the traditional way of developing and selling software. Equilibrium tends to emerge between proprietary software companies. They may compete against one another, but they compete along common grounds; they operate in a similar manner. When something different comes along that is not bound by the same set of rules, the market dynamics can change in any number of ways. In the next section, we explore open source as a market force.

4 FOUR OPEN SOURCE CASE STUDIES

Despite a burgeoning literature on the topic of open source software, there is a lack of research addressing open source software as a market force. In order to derive insight into our particular case of open source CRM software, we make use of concrete cases in which open source projects attempted to enter existing markets.

Four recent examples are considered: Linux operating system, Apache HTTP server, MySQL database server, and PHP scripting language. Our emphasis in each case is on understanding the motivations that prompted the project's creation and development, the market conditions present before its introduction, and its eventual market impacts.

Table 4.1 Overview of Four Case Studies[13]

Case Studies	Description	Competitors[14]
Linux™	Linux is an open source UNIX-like operating system	• Novell • Microsoft Windows • IBM OS/2 • Mac OS
APACHE	Apache is an open source HTTP server used to display webpage content	• Microsoft IIS • Netscape • NCSA HTTPd
MySQL	MySQL is an open source database management system	• IBM DB2 • Microsoft SQL Server • Oracle • Sybase
php	PHP is an open source scripting language used to display dynamic webpage content	• Microsoft ASP • ColdFusion • Java Server Pages (JSP)

4.1 Linux Operating System

The Linux operating system is the most well known open source project. Unlike proprietary competitors Windows and Mac OS, its underlying source code is publicly available. Anyone can use, modify, or redistribute it. Initially developed for and by individual enthusiasts, Linux has since garnered support from the likes of industry heavyweights IBM, Hewlett-Packard, and Novell for use in servers. It is also gaining popularity in the desktop market versus dominant rival Windows.

[13] Republication of the four logos is permitted for non-commercial use under the terms of their respective open source licenses.

[14] Competitors faced at the time of each project's introduction and over the course of its development. Some of the competitors listed are no longer market forces. NCSA HTTPd, for example, was a market leader upon the advent of Apache; it is no longer extant.

In 1983, the GNU Project, founded by Richard Stallman, set out to develop a Unix-like operating system comprised entirely of free software. It succeeded in producing and collecting libraries, compilers, text editors, and a Unix-like shell. The missing ingredient, however, was its kernel.[15] Finnish university student, Linus Torvalds, filled this gap by developing the kernel that would later be dubbed "Linux". The first version of the kernel was released on the Internet in September, 1991.

The market introduction of Linux was met by a wave of both praise and criticism. Supporters emphasized its high degree of reliability and security, while critics focused on its lack of user-friendliness. A broad knowledge of computers was required in order to install and use Linux, which limited its appeal. Subsequent releases, however, have addressed this problem. In fact, the highly regarded software usability consulting firm, Relevantive, concluded that for a set of desktop-related tasks, Linux is now "equal to Windows XP" (Relevantive, 2003). The Linux case thus demonstrates an open source project's ability to enter the market as, in some respects, a poorly performing alternative, and to gradually narrow the performance gap over time.

In terms of its market impact, Linux has not overtaken commercial competitors Windows and Mac OS. Estimated to be in use in only 2.8% of personal computers (IDC study, 2002), it has barely made a dent into Microsoft's dominant market position. Yet, few would disagree that Linux has had a substantial market impact. With servers, in particular, it has succeeded in gaining an estimated 25% of the market (IDC study, 2002). Much of its success in the server market stems from the popularity of the "LAMP" stack.

[15] An operating system's kernel is responsible for providing secure access to the machine's hardware and to its various computer processes.

Comprised of the Linux operating system, Apache web server, MySQL database server, and PHP scripting language, the LAMP stack provides users with everything they need in order to run dynamic web pages – all via open source applications.[16] Its most significant success, however, has been in gaining the support of major commercial IT firms. IBM, Hewlett-Packard, and Novell have endorsed and supported Linux in recent years, even going as far as subsidising part of its development. The question of why profit-driven companies would go to such lengths to support an open source project will be addressed in the next section when we examine the motivations driving open source development.

4.2 Apache HTTP Server

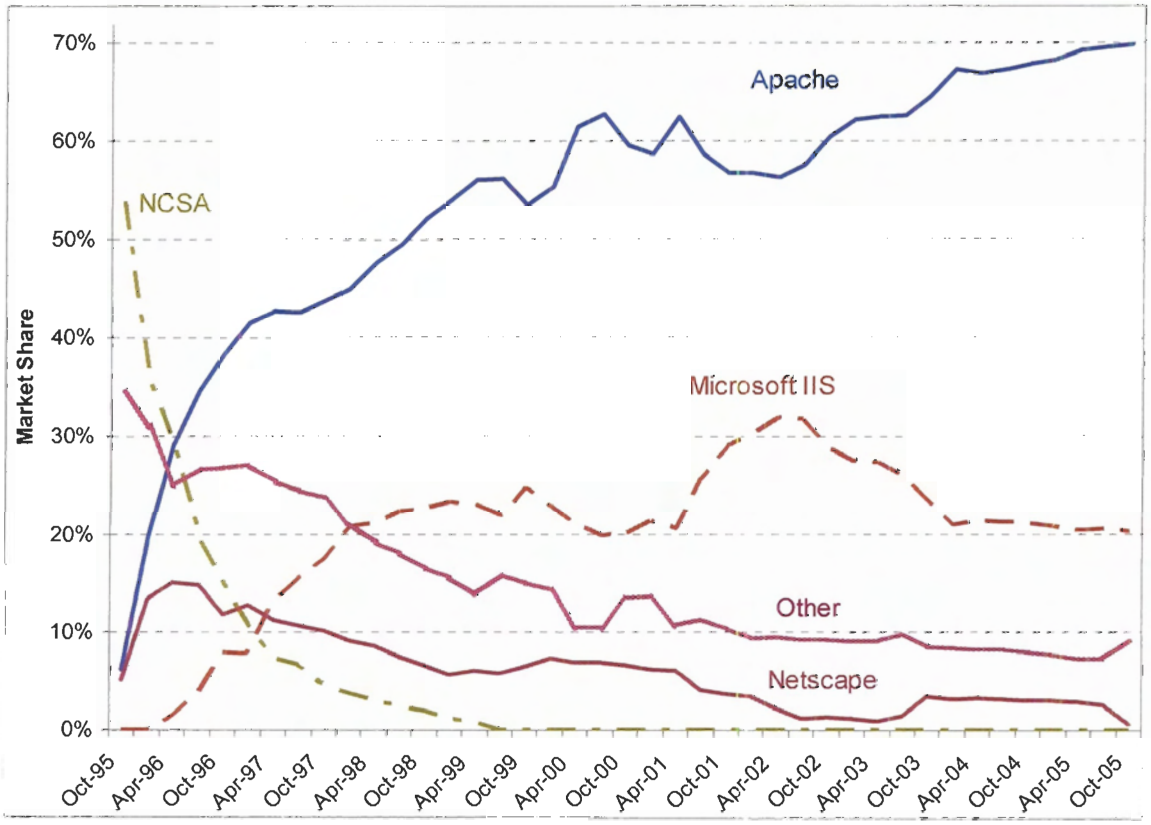
Apache HTTP Server is an open source web server. It, too, forms part of the popular LAMP Stack, comprised of the four applications necessary to operate and manage dynamic web pages: Linux, Apache, MySQL, and PHP. Internet monitoring company Netscraft's August 2005 web server survey estimates that Apache is running on at least 69% of all web servers.

When Apache was first developed and introduced in 1995, its functionality lagged behind that of its main competitors, Netscape and NCSA. Apache's sole advantage was its price. At the time, Netscape and NCSA customers paid little attention to Apache. It was interesting, but not a viable alternative for users requiring advanced functionality. Netscape and NCSA also held advantages in being recognisable and trusted name brands. If customers encountered problems, they would know where to turn. What quality assurances could an unknown open source foundation offer?

[16] The four applications correspond to the four case studies presented in this section.

Given these initial circumstances, it is somewhat surprising Apache succeeded. Apache owes its success to the fact that there was a sizable part of the market that did not require the advanced functionality offered by Netscape and NCSA. For these users, price was the most important factor. In October 1994, Netscape's Communications Server 1.0, for example, was selling for USD 1,495. Its Commerce Server 1.0 was selling for USD 5,000.[17] For a large portion of the market, paying such high prices was not an option. Apache appealed to this non-paying segment of the market, which Netscape and NCSA did not even care about. Thus, Apache's initial gains did not stem from stealing customers away from incumbents, so much as from creating new customers.

Figure 4.1 Apache's Market Share Growth



[17] Netscape's name at the time was Mosaic Communications. They changed it shortly after. An order form is viewable at http://www.dotnetat.net/mozilla/mcom.10.1994/MCOM/ordering_docs/index.html.

For Apache, this non-paying segment of adopters was crucial. It provided an initial community of users, critical to any open source project's development. This group of users tested, modified, and enhanced the product, enabling Apache to quickly improve its offering. Before long, Apache had substantially narrowed its performance gap with respect to Netscape, NCSA, and the other commercial web servers. As a result, more and more commercial customers were willing to consider switching to Apache. This led to an even wider user base, which furthered heightened Apache's rate of improvement. By April 1996, less than a year and a half after its introduction, Apache was already estimated to be the most popular HTTP server in the world.

Apache's rapid success demonstrates that open source projects can have dramatic market impacts – and grave consequences for incumbents. Netscape and NCSA saw their strong market positions collapse in a matter of months. A poorly performing open source project, provided it appeals to an initial core base of users, can use the contributions of the community to improve its offering and quickly move up-market.

4.3 MySQL Database Server

MySQL is a multithreaded, multi-user database management system (DBMS). Unlike the other LAMP applications, MySQL is owned and sponsored by a single for-profit firm, the Swedish company MySQL AB. The company makes MySQL available as open source software under the GNU General Public License (GPL). It also sells MySQL under commercial licensing arrangements for customers that do not wish to incorporate modifications back into the open source community. MySQL AB thus operates as a hybrid model. It supports open source activities, but does so in order to further its for-profit activities.

MySQL was initially ridiculed by database experts for its lack of functionality compared to proprietary relational databases, such as Oracle. MySQL did not support transactions or relational integrity constraints, for example – both standard features of relational DBMS. MySQL AB tried to justify such shortcomings. They claimed to have made a conscious decision to sacrifice functionality in exchange for speed and simplicity. This appealed to a large group of early adopters that did not have the same demanding functionality requirements as Oracle's target customers. Moreover, they could not afford Oracle's advanced functionality. For them, the choice was not between Oracle and MySQL; it was between MySQL and non-consumption. As a result, MySQL quickly grew in popularity. Its sizable user base enabled MySQL AB to gain valuable experience. Each subsequent release of MySQL managed to narrow the performance gap a little more. Features that MySQL AB did not include in early releases were incorporated in subsequent versions. MySQL was then able to appeal to progressively more demanding market segments, gradually stealing market share away from the proprietary relational DBMSs. Although MySQL still falls short of most relational DBMSs in terms of functionality, it offers functionality that is *good enough* to satisfy most mainstream users, and at a lower price. MySQL thus undercuts competitors that offer superior performance, but more performance than most customers need. Such a disconnect between what customers want and what incumbents are offering leaves the door open for new entrants.

4.4 PHP Scripting

PHP is a popular scripting language that allows users to create server-side applications and to display dynamic webpage content. Initially developed in 1994 by Rasmus Lerdorf as a means of collecting data about his personal webpage and displaying his résumé online, Lerdorf soon became aware of its appeal to other users. He published his PHP tools later that year. A 2002 survey by Internet monitoring company, Netscraft, indicates that PHP is the most widely deployed server-side scripting language. Three salient points are of interest: Lerdorf's decision to make PHP an open source project from the beginning, PHP's impact on incumbents, and hybrid business models that have formed in the wake of PHP's success.

As mentioned earlier, Lerdorf created PHP for his own personal use. This is called "consumption utility" by economists. It is a common initial driver behind open source projects, since software users and software developers often overlap. An individual using an application encounters a problem. Perhaps he or she is seeking to accomplish something that commercially accessible products cannot accomplish. The individual proceeds to develop a solution – provided he or she has the skills necessary to do so – only afterwards recognising that others face the same problem, and could make use of the same solution. Such solutions, borne of consumptive utility, are often released freely as open source projects. That the impetus behind the project was a lack of accessible alternatives may have a role in ensuring that the project is made accessible to others.

At the time of PHP's introduction, alternatives were available. They just were not accessible to many given their high price tags. ColdFusion, for example, was developed around the same time as PHP was. Unlike Lerdorf, however, ColdFusion's developer, J.J. Allaire, chose to commercialise his product. ColdFusion, which was later acquired by Macromedia, was once a popular, though expensive, product. It was generally recognised as an effective, easy-to-use, language for writing web-based applications. Yet, while PHP's popularity has skyrocketed, ColdFusion is no longer a serious option for most users. PHP steadily improved its offering, and eventually a tipping point was reached at which ColdFusion could no longer compete. It is understandably difficult to sell a product at a profit when a comparable, free alternative exists. PHP thus overtook ColdFusion and other commercial alternatives.

PHP's success has given rise to an interesting phenomenon. Profit-driven firms have been established, which both contribute to and benefit from the PHP user community. Zend Technologies, for example, is widely recognised as the principal contributor to PHP's development in recent years. The "Zend Engine" is the most popular scripting engine used together with PHP. It is available for download free-of-charge from Zend Technologies' website. The company's revenue model is based on complementary add-on products and services, such as training, certification, and customisation. Zend Technologies thus accumulates reputation and experience through its open source activities to further its for-profit objectives.

The availability of complementary service-oriented firms, such as Zend Technologies, in turn, makes PHP a more viable alternative. It offers potential users contemplating adoption more choices. Open source projects can, therefore, exhibit positive network externalities. As more people adopt PHP, the more valuable PHP becomes to adopters.

4.5 Lessons from Case Studies

These four cases challenge the argument that open source CRM is not a threat because it offers sub par performance. In each case, an open source project entered its respective market as a poorly performing alternative. Yet in each case the open source project succeeded in making a significant market impact. Linux is making substantial inroads into the operating system server market and is now considered by some experts to be on par with Windows XP in usability; Apache is dominating the HTTP server market, leaving commercial competitors struggling to stay afloat; MySQL steadily improved in performance and functionality and now appeals to a wide group of users; and PHP's exponential growth has marginalised commercial competitors, such as ColdFusion.

5 MOTIVATIONS THAT DRIVE OPEN SOURCE

This section is a diversion of sorts. It does not relate directly to our central purpose of evaluating the likely impact of open source CRM. It does, however, serve to place the growing popularity of open source in context. Open source projects are always confronted by a question mark. Why do individuals contribute to them in the first place? Even more puzzling, why do profit-driven companies opt to do so?

5.1 Personal Motivations[18]

From an economics standpoint, the open source development model presents a challenge (Tuomi, 2005). Due to the biases of economic theory in favour of private ownership, monetary reward, and scarce resources, the motivations driving open source development are difficult to explain. Open source developers do not own the goods they produce; the rewards they gain for this development are non-monetary; and, unlike most resources, the open source pool becomes more valuable the more people use it. Given such considerations, how can voluntary investments in open source projects be explained?

An extensive literature seeks to address this question. Hippel (2005) draws attention to non-monetary forms of compensation, such as reputation, experience, and self-fulfillment as significant motivating factors. Lerner and Tirole (2000) emphasize the importance of delayed payoffs, such as enhanced career opportunities and ego gratification stemming from peer recognition. Tuomi (2005) notes that ownership is still

[18] This sub-section is based on the literature review in Tuomi (2005).

an important motivation in open source projects – ownership redefined around control over an evolving good's development, rather than over the good itself. Finally, Hippel and Krogh (2003) argue that developers receive sufficient private benefits from public use of their product – in the form of learning and tailored functionality – to justify continued development efforts. These varied explanations are united by the notion that individual participants in open source projects do receive compensation for their efforts. It is compensation that cannot, however, be measured strictly in terms of dollars and cents.

5.2 Business Motivations

There is another set of motivations that have not drawn near as much attention in the research literature. What prompts profit-driven companies, under certain circumstances, to support open source projects? The willingness of corporations to do so, both in terms of development and of adoption, is of paramount importance. It is what transforms seemingly harmless open source projects into significant market forces. Profit-driven firms do not support open source on ideological grounds. Their actions stem purely from selfish considerations – what we may call "enlightened self-interest". Here we seek to elaborate upon the different forms of enlightened self-interest that serve to align commercial interests with those of open source.

Table 5.1 Summary of Business Motivations

Motivations	Objectives	Examples
To Establish an Industry Standard	• Jumpstart technology adoption • Create a self-reinforcing virtuous cycle	• IBM & Eclipse • HP eSpeak
To Change Supplier Relationship Dynamics	• Find alternatives to dominant supplier • Shift the balance of power towards customers	• Support for Linux versus Windows
To Stifle Innovation in an Emergent Field	• Prevent commercial development of new inventions • Remove incentives for others to innovate	• Kodak's sponsoring of digital photography research
To Complement Wider Strategic Interests	• Advance broader strategic goals • Create synergies with core business activities	• IBM Global Services • Hewlett Packard
To Avoid Being Back into a Corner	• Re-orient business strategy • Flip tables on commercial competitors	• Borland's Interbase
To Take Down an Industry "Gorilla"	• Use open source initiative to mobilise widespread support against a dominant competitor	• Widespread support of Eclipse versus MS Visual Studio
To Unload an Aging Product Line	• Avoid the need to support dying product lines indefinitely • Utilise open source community as a means of outsourcing support functions	• Netscape's web browser • SAP's relational database
To Gain Reputation and Credibility	• Leverage experience, expertise, and reputation gained with open source activities to further commercial interests	• Zend Technologies & PHP • MySQL AB

5.2.1 To Establish an Industry Standard

A complex, poorly understood sequence of events has to occur in order for a product or technology to take off and become a standard. Enthusiastic developers are needed to produce content that will attract end users. An established user base is needed to convince developers to produce content. Positive events on either side of the equation can trigger a virtuous cycle of adoption. Setbacks can lead to a vicious cycle of non-adoption. Making the product open source gives both sides a little extra push towards adoption. Doing so builds trust and reduces suspicion in the developer community. For end users that might otherwise wait for the product to establish itself before purchasing, being able to sample it for free encourages earlier adoption. Making a product open source certainly does not guarantee widespread adoption. Doing so can, however, jumpstart the process by encouraging developers and users to come on board earlier.

5.2.2 To Change Supplier Relationship Dynamics

In one-sided CRM supplier-customer relationships, customers can potentially look to open source as an alternative. Much of the support behind the Linux operating system stems from a practical desire for an alternative to Microsoft and Windows. With no alternative to choose from, Microsoft dominates the relationship. Microsoft's customers thus have an interest in seeing a viable open source competitor gather momentum. Even for customers that have no plans of actually switching, the mere fact that they *could* switch serves to empower them. The lengths firms would go to actively support an open source alternative for such reasons depends on their level of dissatisfaction with current suppliers.

5.2.3 To Stifle Innovation in an Emergent Field

This may seem counterintuitive. Why would a company support open source in order to stifle innovation? Why would a company want to stifle innovation in the first place? Agrawal (2002) notes that Kodak has sponsored research at M.I.T. in areas related to digital photography – a curious move on their part, but certainly nothing particularly shocking. What is surprising is that the sponsorship stipulates no exclusive licensing. In other words, any intellectual property (IP) derived from the research has to be released to the public domain. Historically, sponsoring firms usually win the rights to IP they sponsored (Agrawal, 2002). Moreover, exclusive licensing has been the norm at universities since the Bayh-Dole Act of 1980. Thus, even though Kodak would be a prime candidate to license such IP themselves, they chose to explicitly disallow exclusive licensing.

Trying to discern why, Agrawal (2002) puts forth an interesting hypothesis: firms sometimes use non-exclusive licensing "in order to prevent, or at least slow down, the commercial development of inventions in a particular technological market" (Agrawal, 2002). Kodak was concerned the emerging field of digital photography would negatively impact their core line of 35mm film products. Sponsoring research in digital photography that would then be made public was, therefore, an attempt to eliminate the incentives to develop and commercialise innovations in digital photography. With the latest innovations publicly available, others would have less incentive to invest their resources in digital photography research. With no profit to be made, there would be less incentive to innovate. This represents a different slant on business support for open source: not trying to profit from open source oneself, but using it as a means of preventing others from profiting from a new technology perceived as being in some way threatening.

5.2.4 To Complement Wider Strategic Interests

In 1999, IBM produced a software development framework called Eclipse. In 2001, after having spent an estimated $40 Million on its development, IBM opted to release Eclipse to the open source community (JavaWorld.com, 2002). Part of IBM's rationale for doing so was to complement its strategic re-orientation towards consulting services. More generally, companies that focus their strategies around services, such as consulting, customisation, and training, are more likely to support open source. If a firm's core activity is not selling software, but rather accompanying services, the open source approach makes a lot of sense. With a hand in open source development projects, service firms can leverage their expertise and reputation in order to gain customer confidence. Furthermore, open source based implementation proposals are inherently more reassuring for customers that might otherwise be concerned about being locked-in – and potentially stranded – with a particular vendor's proprietary standard. Projects based on an open source platform afford customers other options if, for example, the implementing firm goes out of business. Thus, companies can support open source projects when doing so complements their overriding strategic objectives, which is often the case with service-based firms.

5.2.5 To Avoid Being Backed into a Corner

Profitable incumbents, such as ColdFusion, have a lot to lose from open source projects, such as PHP. An incumbent that is not profitable, however, does not. A commercial software company losing market share to other commercial software companies could begin to feel backed into a corner. A tipping point is reached at which supporting open source begins to seem like the only way out. Releasing a product to the open source community may be an attractive option. Doing so can augment one's reputation in the open source community, allowing for a re-orientation of company strategy around services instead of products. Meanwhile, former competitors would have to contend with the rising tide of open source, which would have presumably received a functionality boost from the company's release of source code. As a move of desperation, supporting open source can therefore be a viable option.

5.2.6 To Take Down an Industry "Gorilla"

Another motivation, related to the previous one, could arise if a group of competitors recognises that they are losing market share to one large competitor, which is fast becoming dominant. Such was the case with IBM and Eclipse. IBM realised that MS Visual Studio was becoming the standard. By making Eclipse an open source project, IBM was able to enlist the support of former competitors Borland and WebGain, among others, in order to counter the emerging market gorilla.

It may be argued that such an alliance need not have been behind an open source project. They could have instead created a strategic partnership to develop an alternative to Microsoft's Visual Studio, and kept the shared product proprietary. Making a product open source, however, makes it easier to mobilise support behind one alternative. Had Eclipse remained "IBM Eclipse," it is unlikely that IBM's chief competitors would have been as willing to back it.

5.2.7 To Unload an Aging Product Line

A recurring problem software companies encounter is having to support aging product lines indefinitely. When a company decides to "kill" off an old product, customers still using it feel betrayed and abandoned. Customers understandably expect that purchased products will continue to be supported and maintained for as long as they need. For growing companies, however, this is impractical. Software development companies can unintentionally morph into bloated support organisations, consumed with servicing customers from many generations of past product lines, all the while neglecting their core activity of developing new software.

Open source communities are increasingly seen as an escape from this conundrum. Releasing an aging product line to the public domain allows it to die a more natural death. As long as the product still has users, there will be support options available from within the community. Once the product's user group shrinks to nothing, the product can officially be considered "dead", without risking any consumer backlash.

5.2.8 To Gain Reputation and Credibility

As noted above, individual programmers often contribute to open source projects for non-monetary benefits, such as reputation, credibility, and experience. These non-monetary forms of compensation motivate commercial companies in much the same way. This applies especially to products for which gaining support amongst developers is a factor critical to adoption success. Open source contributions can be an effective means of gaining credibility and of tapping into the developer community.

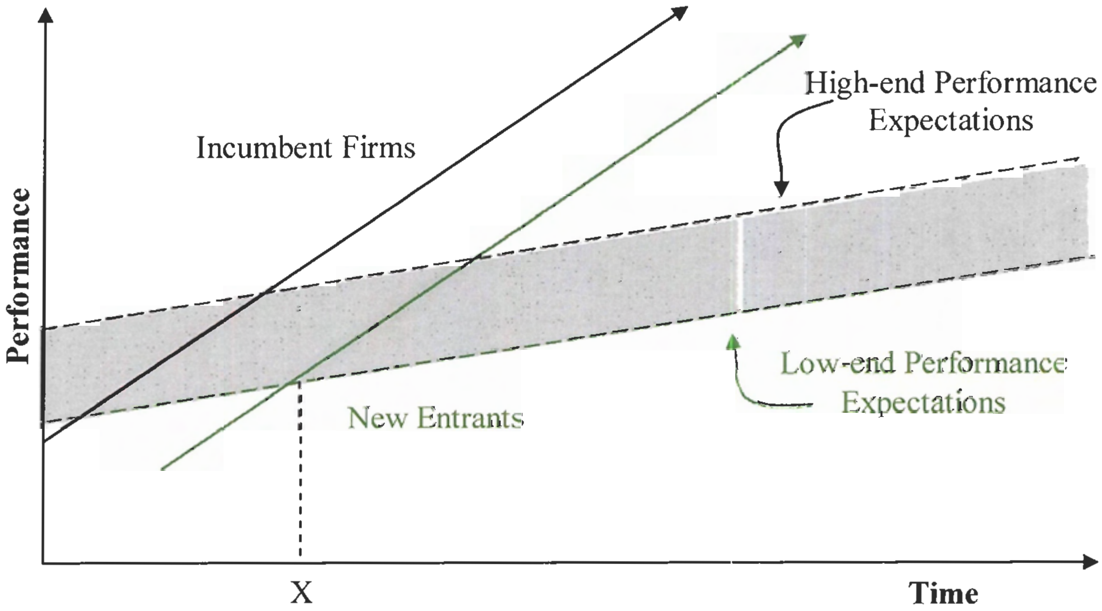
6 OPEN SOURCE AS A DISRUPTIVE TECHNOLOGY

A clear pattern can be discerned in the cases presented above. In each case, an open source project entered its respective market as a poorly performing alternative to commercially available products. Linux was difficult to install and operate; Apache was "a patchy" imitation of Netscape; MySQL was ridiculed by database industry experts for its lack of functionality; and PHP did not offer the service, reliability, or performance that ColdFusion did. Yet in all four cases, they succeeded in making inroads into established markets, in some cases even supplanting their superior performing competitors. The obvious question is "how?"

Clayton Christensen's model of "Disruptive Technologies" provides an intuitive explanation of these open source success stories. Christensen coined the term to refer to a new technological innovation, or business model that overturns an established market, despite offering initially substandard performance. It forms a subset of Porter's well-documented notion of new entrants acting as substitutes in a given market.

Christensen's premise is simple. The pace of technological progress sometimes exceeds the performance improvement customers are either willing or able to absorb. As a result, market incumbents sometimes overshoot customer performance expectations, leaving the door open for new entrants to undercut them. In every industry, there are several product dimensions along which mainstream customers measure performance. For each dimension, the performance of the new entrant can be inferior to that of incumbents, as long as it is *good enough* for less demanding market segments. By appealing to less demanding users, the new entrant can establish a foothold, improve its offering, and gradually narrow its performance gap with incumbents. As it does so, it appeals to progressively more demanding segments of customers, previously beyond reach. At time X in *Figure 6.1,* for example, the new entrant is just starting to gain traction with customers on the low end of the performance demand scale. Their demands are being met by the new entrant, and overshot by incumbents – meaning that incumbents are offering more performance than they need or want.

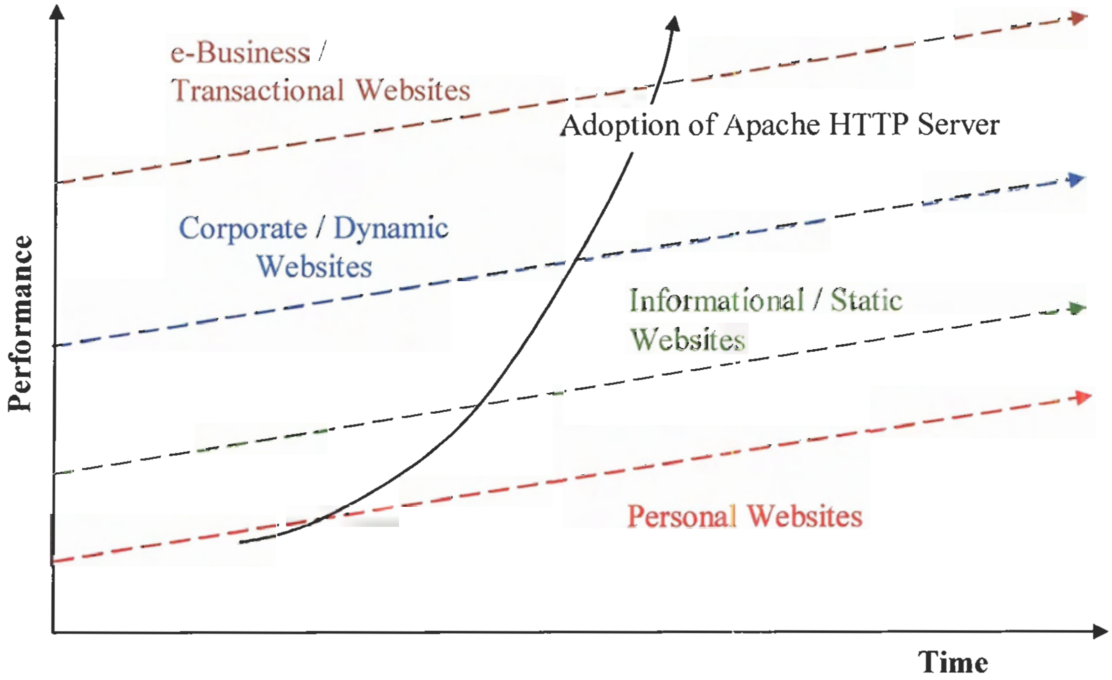
Figure 6.1 New Market Entrants Undercutting Incumbents


Incumbent Firms
High-end Performance Expectations
Performance
Low-end Performance Expectations
New Entrants
X
Time

Source: Based on Christensen, 1997

Take the case of Apache HTTP server. Initially, it offered performance, reliability, and support far inferior to those of commercial servers, such as Netscape and NCSA. But it was less expensive. Although, the higher demanding segments, such as large e-commerce firms had no interest in switching, Apache appealed to lesser demanding segments, such as individual webpage designers. Over time, Apache's quality improved. It could then compete in progressively more demanding segments, eventually displacing incumbents.

Figure 6.2 Adoption of Apache along different Market Segments


e-Business / Transactional Websites
Adoption of Apache HTTP Server
Corporate / Dynamic Websites
Informational / Static Websites
Personal Websites
Performance
Time

Source: Based on Christensen, 1997

Whether open source CRM software fits Christensen's model of a disruptive technology will be important in assessing its likely long-term impact.

7 COULD OPEN SOURCE CRM BE DISRUPTIVE?

Christensen's model of disruptive technologies goes a long way towards explaining the successful market entries of our four case studies. Could it be applied to open source CRM too? In this section, we compare open source CRM with the four cases of Linux, Apache, MySQL, and PHP. Doing so will enable us to determine the extent to which the lessons derived from these four cases can be applied to open source CRM.

7.1 Similarities

There are clear similarities between open source CRM and the four cases we examined earlier. Open source CRM products, such as SugarCRM, have entered the market as poorly performing alternatives relative to commercial competitors. In the CRM industry, there is also a large segment of non-users that an open source product could potentially establish a foothold with. Moreover, the CRM industry is highly fragmented, as was the case with the four LAMP applications.[19] A final, less obvious similarity is the ability to offer customers a "whole product solution". The LAMP applications succeeded in part because together they provided users with everything they needed to operate dynamic web pages. This is true of open source CRM as well. CRM software requires a database server, for example. For commercial CRM customers, this means having to purchase a compatible server, such as Microsoft IIS. Open source CRM software, on the other hand, can build upon the freely available LAMP foundation.

[19] Excepting Linux in the desktop market.

7.2 Differences

There is one important difference that calls into question the suitability of these four cases as reference points. The LAMP applications are all tied to the Internet. As such, they each had large potential user bases, with low performance expectations, to draw upon. More importantly, the fact that they related to the Internet meant that the developer and user communities almost completely overlapped. Individual programmers developed code for their own utility, only afterwards contributing to the larger community – and vice versa. In the CRM industry, it is unlikely that the same people developing code are going to be the ones actually using it. Conversely, the people using the applications are unlikely to possess the skills necessary to code them themselves.

7.3 Criteria for Open Source CRM to be Disruptive

Despite this notable difference, open source CRM closely resembles Linux, Apache, MySQL, and PHP. In order for open source CRM to emerge as a viable alternative, however, two conditions – critical in each of the four cases – would need to be met: (1) there has to be sufficient reason for switching; and (2) open source CRM has to narrow its performance gap with commercial incumbents. The next two sections explore the CRM industry to see whether there is sufficient impetus for change. *Section 10* then proceeds to outline events that would narrow the performance gap between open source and commercial CRM. By evaluating their likelihood of occurring, we will be in a better position to assess the threat open source CRM poses to Pivotal.

8 CUSTOMER CONCERNS WITH THE CRM INDUSTRY

For open source CRM to gain traction with customers in the CRM industry, customers must have sufficient reason to switch. The success of Linux, Apache, MySQL, and PHP was due in large part to segments of customers not being satisfied with commercial alternatives. From a customer's perspective, there are three fundamental concerns with commercial CRM vendors: (1) extreme vendor lock-in; (2) unfavourable payment schedules; and (3) a lack of available customisation options.

8.1 Extreme Vendor Lock-in

Once a CRM customer commits to a particular vendor, the costs associated with later switching to a different vendor are prohibitive – the customer is said to be "locked-in". The high switching costs are due to several factors. Purchasing CRM software almost always involves substantial sunk cost investments. Different vendors make use of different data models, which leads to a lack of interoperability between proprietary CRM platforms. Switching between them is costly and often impractical. As a result, initial investments cannot be recuperated if a customer later decides to switch. In addition to financial investments, significant investments in time and effort are required to integrate, implement, and learn to use new CRM software. The natural tendency, therefore, is to stick with the same vendor. Unless customers are extremely unsatisfied with their current CRM supplier, the costs of switching far outweigh the benefits.

Such an arrangement is beneficial to suppliers, but problematic for customers. It limits a customer's ability to exert any degree of control over the relationship. Even if a customer is satisfied and has no intention of switching, being able to do so should the need arise is important. Being locked-in to a particular supplier, limits a customer's flexibility and freedom of action.

8.2 Unfavourable Payment Schedules

Another concern with the CRM industry is the revenue model that commercial CRM providers operate under. The emphasis is tilted towards large, upfront license fees. The degree to which this revenue model is problematic depends on the customer. For larger customers – the "Fortune 500" market segment – it is not as troubling. For the SMB segment, however, license fees represent a major hurdle in the way of adoption. SMB customers simply cannot afford substantial upfront costs. Most SMB "customers", therefore, fall into another, larger segment of the market: non-users.

8.3 Lack of Customisation Options

CRM vendors prefer to sell pre-packaged, standardised products. SAP, for instance, is renowned for its uncompromising stance on standardisation. Customers are confronted with a dilemma: "Our way or the SAP way." Customisation is possible via add-on services and charges, but the product itself is standard.

CRM software addresses a core activity: managing customer relationships. To what extent is managing customer relationships a common process across different organisations and industries? In other words: are the needs of CRM customers homogeneous or heterogeneous? Certainly, there is much that is common across organisations and industries. Almost every company needs to store customer contact information for multiple communication channels, for example. There is also much that differs. The way companies relate to their customers is often tied to their corporate culture. There are idiosyncrasies there that may very well be integral to the firm's operations. Standardising such uniqueness away is not always desirable. Most CRM purchases, therefore, require at least some degree of customisation. A customer's ability to customise commercial CRM software, however, is limited. While commercial vendors offer to perform such customisation themselves, the fees charged for doing so are usually prohibitive. That the software's code is closed restricts the availability of third-party firms that can provide customisation services. A customer's selection is therefore limited.

8.4 Vendor-Customer Tradeoffs

These three "concerns" are not concerns for everyone. For commercial CRM providers, vendor lock-in ensures favourable customer relationships. A basic rule of marketing is that if your product does not have inherent switching costs, it is best to artificially create them – via loyalty programmes, for example. Thus, although lock-in is a concern for CRM customers, it is extremely beneficial for CRM vendors. They have every reason to prevent this "problem" from ever being solved. Likewise, the unfavourable payment schedules are only unfavourable for customers. Vendors prefer that customers pay large upfront fees. As long as customers are willing to do so, there is no reason for vendors to fix this "problem" either. The same applies to the third "concern" of a lack of customisation options. While customers prefer greater customisation, vendors prefer standardisation. The three areas of concern thus involve tradeoffs between CRM vendors and CRM customers. The question is: Would customers continue to accept extreme vendor lock-in, large upfront fees, and highly standardised products if there was a viable alternative available?

9 RECENT DEVELOPMENTS IN THE CRM INDUSTRY

Interpreted in light of the three problems presented in the previous section, recent developments in the CRM industry point to where the industry may be headed.

9.1 The Success of Salesforce.com

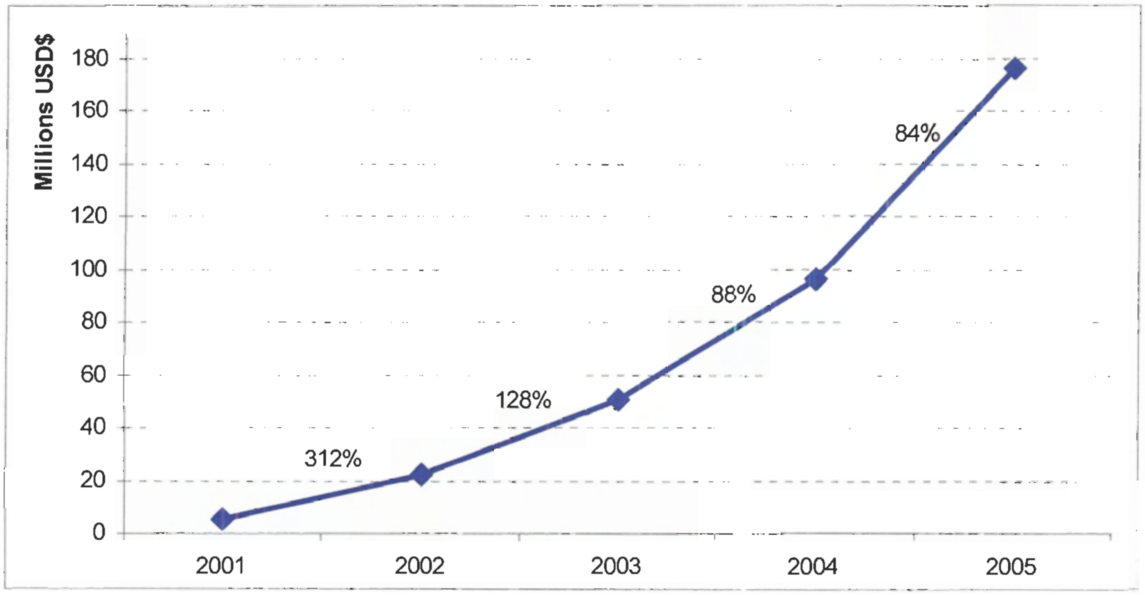
As mentioned earlier, there has been a trend in recent years, epitomised by the success of Salesforce.com, towards hosted CRM solutions. The Salesforce.com approach offers several advantages. It eliminates the need for large upfront fees. Customers instead pay smaller monthly fees.[20] It also requires far less commitment from customers. Whereas purchasing and implementing SAP or Oracle's CRM solutions constitutes a massive undertaking, Salesforce.com can easily be sampled. Furthermore, the burden of maintenance is to a large extent shifted away from the customer to the vendor. Since Salesforce.com hosts the product themselves, maintaining the system is their concern. Two disadvantages characterise such an arrangement. First, customers relinquish a degree of control over their potentially sensitive data. Doing so can raise troubling questions of accountability. What happens if Salesforce.com's system is compromised or hacked into? The subscription agreement absolves Salesforce.com of responsibility, placing the burden of absorbing such costs squarely on the customer. Second, by hosting customers' CRM data, Salesforce.com makes customers dependent on them. Effectively, customers outsource their IT assets to Salesforce.com, which leads to organisational "de-skilling".

[20] That is not to say that hosted solutions are more economical. A comparison of the total costs of ownership over a three year period between Salesforce.com and client-based CRM vendors, for example, shows that Salesforce.com is actually more costly in the long run (see *Appendix A*). The point here is that Salesforce.com offers customers more favourable cash flow.

Customers never own the system and become highly dependent on Salesforce.com to continue spoon feeding them CRM. Switching costs are therefore extremely high. Salesforce.com offers customers the convenience of paying for CRM services monthly, but once that decision has been made, they must continue to pay indefinitely for continued use.

Salesforce.com's novel approach to CRM has had a dramatic impact on the CRM industry. It has appealed to a large portion of the market. Customers that find unfavourable cash flows more troubling than the accountability and dependency drawbacks have embraced hosted CRM solutions.

Figure 9.1 Salesforce.com's Revenue Growth



Framed in terms of the three fundamental problems discussed above, the success of Salesforce.com makes sense. Salesforce.com mitigated the revenue model problem by spreading costs over monthly payments. In doing so, however, it further aggravated the lock-in problem. To whom does this trade-off appeal? It appeals primarily to SMBs for which gaining financial flexibility is worth sacrificing informational flexibility.

9.2 The Emergence of Open Source CRM

That brings us to the threat of open source CRM. How does open source CRM fit into the industry? We have seen the three problems that figure prominently in customers' adoption decisions. Open source CRM holds the potential to remedy all three. Lock-in is far less severe. Given that customers have access to the underlying source code, it is not nearly as difficult to switch to another third-party service provider. Without the need for large upfront license fees, open source CRM's revenue model is more flexible. It is better suited for financially constrained customers, which applies to virtually the whole SMB segment. With the focus shifted from providing pre-packaged products to custom services, open source CRM is also better equipped to deal with the inherent heterogeneity of the CRM market. It thus stands to reason that customers would have sufficient motivation to switch if a viable alternative existed.

10 EVENTS THAT WOULD NARROW PERFORMANCE GAP

What would make open source CRM a viable alternative? We noted earlier that there is currently a performance gap between the offerings of open source CRM and those of commercial vendors. In order to assess the likelihood of open source CRM disrupting the CRM market, several future events are presented that would narrow this performance gap – in line with Christensen's thinking on disruptive technologies.

Table 10.1 Summary of Future Events

Events	Triggers	Impacts
Open Source Creates a New Market Segment	• Customers on the low end of the performance expectation scale are willing to forsake performance for price	• Increases open source CRM's user base • Provides greater resources for feedback, modifications, and improvements
Commercial CRM Customers Backwards Integrate	• Commercial CRM customers become frustrated and/or dissatisfied with commercial vendors	• Boosts credibility of open source alternative • Provides increased funding and resources for open source CRM
Complementors Emerge in Open Source CRM	• User base becomes sufficiently large to attract open source CRM service firms	• Provides potential adopters with more customisation options • Reduces risks associated with switching to open source
Commercial CRM Vendor Switches to Open Source	• Commercial vendor loses market share and can no longer compete	• Provides source code of commercial product to open source community • Gives open source CRM a functionality boost
Open Source CRM Gains Support from a Reputable IT Firm	• Possibility of gaining reputation and credibility by associating name with successful open source project attracts support	• Increases reputation and credibility of open source CRM • Reduces risks associated with switching to open source

10.1 Open Source CRM Creates a New Market Segment

One possible scenario would be for an open source provider, most likely SugarCRM, to gain traction with a segment of customers willing to forsake the greater capabilities of commercial CRM for the cost savings of open source. Attaining a sizable user base would have salutary effects. User contributions and feedback would supply the research and development for future releases. Even users that make use of code from the community without contributing anything back (i.e. free-riders) are beneficial. By using the contributions of others, they are still serving as testers (even if they do not realise it).

Several market segments could potentially fill this role, price sensitivity being the common denominator between them. Non-profit organisations, for example, could benefit from implementing CRM software. The costs of commercial CRM software, however, are beyond their limited budgets. The comparison, therefore, would not be between the performance of commercial CRM and the performance of open source CRM – it would be between open source CRM and nothing. For non-profit organisations and other non-user groups, the current limitations of open source CRM software would not be of primary concern.

The experience, expertise, and modifications derived from such an early community of users would serve to narrow the performance gap. The software would then appeal to slightly more demanding customer segments. The expanded user base would spark further performance enhancements, which would, in turn, enable open source CRM to appeal to progressively more demanding market segments – continuing the cycle.

10.2 Commercial CRM Customers Backwards Integrate

Commercial CRM customers have good reason to at least keep an eye on developments in open source CRM. Should a sustainable open source alternative emerge, it would help them in two ways. First, they could derive benefits from switching to an open source provider themselves. Doing so would eliminate license fees, reduce lock-in, and offer greater potential for customisation. Second, even without switching, having a viable alternative would afford them greater leverage in their relationships with CRM vendors.

Given the interests customers have in seeing a viable open source alternative develop, one or more customers may play a role in contributing to such an eventuality. The level of contribution could vary. On one extreme, a group of commercial CRM customers could join forces, forming a consortium of sorts, to actively support open source CRM. The other, less involved extreme, could entail providing financial contributions without actively supporting it. In either case, it is clear that commercial CRM customers could represent a powerful force in making open source CRM disruptive.

10.3 Complementors Emerge in Open Source CRM

A factor currently hindering the adoption of open source CRM is the difficulty in putting it to good use. Most users do not have the in-house expertise necessary to do so. Anyone can download SugarCRM, for example, but to actually use it to address specific business needs requires outside help. SugarCRM provides customisation and consulting services for a fee, but they have yet to establish a brand name that customers can trust. Moreover, a selection of reputable service firms to choose from is important in order to avoid being locked-in with any one firm. Once such a selection is available, open source CRM will pose a much greater threat to commercial vendors.

The creation of open source CRM-based service firms is conditioned by opportunity. If there are enough users in need of such services, profit-driven companies will surface to address that need. There is nothing standing in the way of new entrants. Entry barriers are low. Furthermore, the fact that the software is open source makes it especially easy for new entrants to get their feet in the door.

10.4 A Commercial CRM Vendor Switches to Open Source

This scenario stems from two of the business drivers outlined in *Section 5:* (1) to avoid being backed into a corner; and (2) to complement wider strategic interests. As noted earlier, the CRM industry is highly fragmented. Should an existing commercial CRM vendor sense that it is losing market share and cannot compete, the possibility of re-orienting its strategy around services and releasing its product to the open source community becomes very real.

This sense of being backed into a corner could be prompted by losing market share to other commercial vendors. It could also perhaps be prompted by open source CRM making inroads into its customer base. By making its software open source, the company could leverage its expertise in order to sell consulting and other complementary services, much like IBM Global Services does.

The benefits to open source CRM are clear. The releasing of a commercial CRM product to the open source community would provide a boost in functionality and expertise that would further narrow the performance gap between commercial and open source CRM.

10.5 A Reputable IT Firm Endorses and Supports Open Source CRM

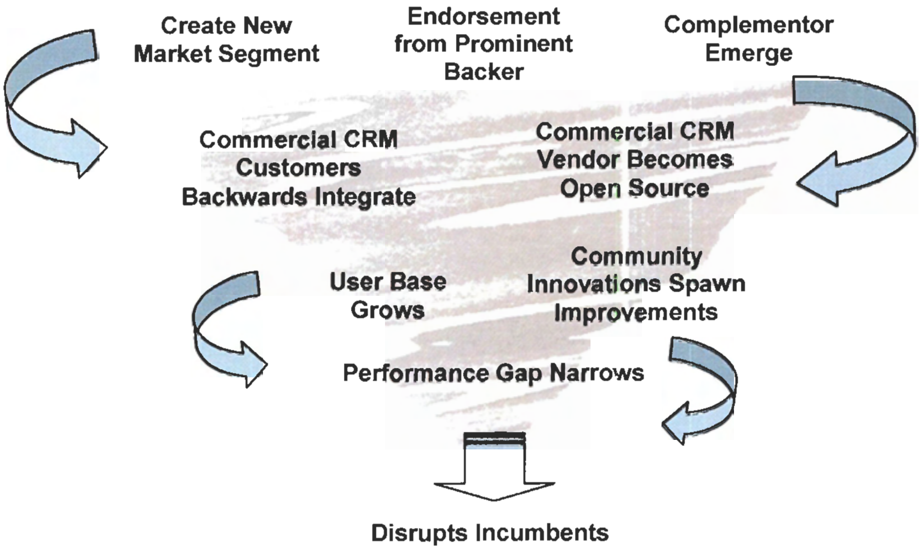
The success of the Linux operating system owes much to the support it received from Red Hat and, later, from IBM. Open source projects always face a question mark regarding their sustainability. Companies are reluctant to adopt them until they have proven that they are there to stay. An endorsement and funding from a reputable industry name could go a long way towards allaying such concerns.

What type of company would be a likely candidate to endorse open source CRM? As in the last scenario, an unsuccessful commercial CRM vendor would be one possibility. Unprofitable as a commercial vendor, it could decide to re-orient its strategy around complementary services and endorse an open source project. Perhaps more compelling would be for a large IT consulting firm to back open source CRM. A prime candidate would be a company often mentioned in connection with open source projects, IBM – and, in particular, IBM Global Services. IBM already has a hand in the CRM industry, offering customisation services for SAP implementation projects. IBM would clearly stand to gain from a successful open source CRM project closely associated with its name. The prospect of an IBM endorsement at some point, therefore, seems likely.

10.6 Cycle of Disruption

These events are by no means mutually exclusive. On the contrary, they are interrelated and self-reinforcing. An endorsement from a reputable IT services firm such as IBM, for example, would improve open source CRM's ability to establish a market foothold. This initial segment would have a demand for customised services. Third-party open source CRM service firms would likely emerge to address this demand, which would, in turn, make open source CRM a more viable alternative for commercial CRM customers.

Figure 10.1 Cycle of Events Leading to Disruption


Create New Market Segment
Endorsement from Prominent Backer
Complementor Emerge
Commercial CRM Customers Backwards Integrate
Commercial CRM Vendor Becomes Open Source
User Base Grows
Community Innovations Spawn Improvements
Performance Gap Narrows
Disrupts Incumbents

It is, of course, difficult to state with any certainly the chronological order of events. The different scenarios influence each another, creating a kind of domino effect. The question is: Which domino would likely be the first to fall? As always, the advantage lies on the side of the status quo. Supporting an emerging project early on is risky – it could still fail. Once it succeeds in garnering that difficult-to-come-by initial support, however, others are usually quick to jump on board. As we have shown, this would lead to an increased open source CRM user base, which would, in turn, narrow its performance gap with commercial CRM products. Given the advantages offered to customers in terms of the three problems discussed earlier, a comparable open source CRM offering would disrupt incumbents.

10.7 Assessment

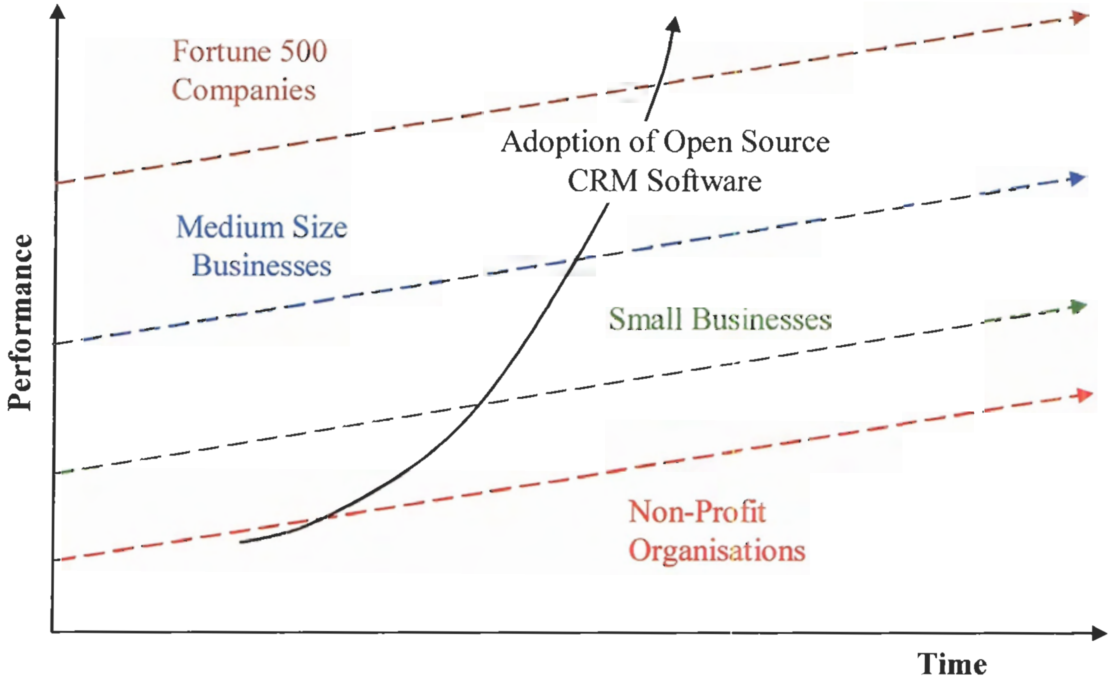
Trying to ascertain what the future holds in store is no easy task. There is a fine line between the industry analysts of today and the oracles and prophets of old. We have presented a vision of what our analyses suggest will transpire in the CRM industry. How likely is it to happen? What level of confidence can we attach to our findings?

Although there are forces working for and against a shift towards open source CRM, the fundamental problems that presently confront commercial CRM customers are likely to push such a shift forward. These five intertwined events make for a robust path to market disruption. Should any one event materialise, it would give impetus to the other four, and so forth. Our conclusion, therefore, is that open source CRM is likely to disrupt commercial CRM vendors. Based on the analogies represented in other disruptive open source projects, a timeframe of five years or less seems probable.

10.8 Implications

This conclusion carries significant implications for commercial CRM vendors. If a comparable open source alternative emerges, commercial vendors can no longer expect customers to pay substantial license fees. The commercial CRM revenue model will not be sustainable in such an environment. Companies that fail to adapt their strategies to the changing circumstances will be left behind – much like ColdFusion was left behind in the wake of PHP's success.

Figure 10.2 Open Source CRM's Likely Adoption Path



Source: Based on Christensen, 1997

The threat is not equal for all commercial CRM vendors, however. The first strategic group, consisting of the "big three" vendors, is far less vulnerable to competition from open source. Given that the customers most attracted to an open source alternative are likely to be those at the low end of the "performance demanded" range, the vendors targeting the SMB segment would be the first to be disrupted (see *Figure 10.2*). They should also be the first to act upon the threat.

11 IS OPEN SOURCE CRM A THREAT TO PIVOTAL?

This has been our central question. In order to answer it, we introduced four open source projects that succeeded in disrupting incumbents in their respective industries. Christensen's model of disruptive technologies provides insight into the reasons behind their successes. Although there are some notable differences between open source CRM and the open source case studies presented, they are very similar. The factors present in each of the four cases are present in the case of open source CRM as well. It therefore stands to reason that open source CRM *could* follow a similar path in disrupting CRM incumbents.

To determine the likelihood of such an occurrence, we directed our attention to the CRM industry. From the perspective of commercial CRM customers, three problems were noted that could prompt wide-scale changes: (1) extreme vendor lock-in; (2) unfavourable payment schedules; and (3) the need for greater customisation. The recent success of hosted CRM vendors, such as Salesforce.com, is easy to understand in relation to these problems. Salesforce.com offered customers a more favourable payment schedule, but did so at the expense of even greater vendor lock-in. Open source CRM could, potentially at least, offer customers a fundamentally different solution – one that addresses all three customer problems.

Taking the analysis a step further, we examined future events that would narrow the performance gap between open source and commercial CRM software. These interrelated events could trigger a chain reaction, increasing open source CRM's popularity and rate of improvement in a self-reinforcing cycle. A reduced performance gap, coupled with open source CRM's diminished lock-in, lower costs, and greater customisation possibilities, would present a fundamental threat to commercial incumbents.

In the final analysis, what conclusions can be drawn? We have shown that open source CRM could very well disrupt the CRM market. We have presented a realistic picture, based on the successes of other open source projects, of how it *could* happen. We have not proven that it *will* happen. The leap between can and will is somewhat tenuous. We have, however, shown that its occurrence is sufficiently likely to warrant serious consideration from commercial CRM vendors.

12 PIVOTAL'S STRATEGIC OPTIONS

The preceding analyses have shown that it is possible, and even likely, that open source CRM will disrupt the CRM market. Moreover, we have concluded that the commercial vendors most likely to feel the sting from a disruptive open source alternative, at least initially, are those targeting the SMB market segment. If these conclusions have merit, then Pivotal is in a perilous position.

In this section, we outline some of the options Pivotal could reasonably pursue in order to better position itself against the threat from open source CRM.

Table 12.1 Summary of Strategic Options

Option	Description	Purpose
Maintain Status Quo	• Maintain focus on commercial product	• "Milk" existing customer base
Adopt a Full Open Source Strategy	• Release source code to open source community • Re-orient strategy around providing custom CRM services	• Become pioneer / leader in emerging open source CRM alternative
Foster a Community of Pivotal Users	• Encourage feedback, modifications, custom projects • Serve as "innovation arbitrator"	• Harness community interaction to improve product offering
Emphasize Non-License Based Revenue Streams	• Place greater emphasis on maintenance, customisation, training, and other non-license based revenue streams	• Reduce dependence on licence fees
Support Open Source Side Project	• Create and support a parallel open source project	• Gain the benefits of open source; still maintain proprietary product
Create Scaled-Down Product Line	• Create a watered-down Pivotal product that could be sold at reduced price	• Appeal to price-sensitive non-users without relinquishing proprietary software model

12.1 Maintain Status Quo

One option would be to simply hang on. This is more than just a token option thrown in for the sake of completeness. Even if open source CRM does succeed in disrupting the CRM industry, Pivotal could reasonably expect to be able to "milk" its existing customer base for quite some time – perhaps indefinitely. Given the high degree of vendor lock-in, some customers may be reluctant to switch, even with a better alternative available. The competitive landscape would not be the same, though. Attracting new customers would be far more challenging with a comparable open source alternative on the market. Still, this option offers simplicity and convenience. Moreover, it would not involve having to take any bold risks – although sometimes not taking risks can be the riskiest option of all.

12.2 Adopt a Full Open Source Strategy

The opposite extreme would be for Pivotal to release its software to the open source community and re-orient its strategy around services. Revenue streams centered on customisation, training, and complementary add-on products would enable Pivotal to leverage its expertise and reputation to position itself as a CRM service leader. This option is confronted by questions of internal feasibility, however. It would be difficult to implement such a drastic change in strategy. Moreover, even if such a feat were accomplished, it would carry substantial risks. What if our earlier conclusions did not pan out? Pivotal's shift to open source would, ironically, be contributing to the open source movement that prompted its actions in the first place. Such a bold move could offer certain advantages, though. Should a disruptive open source alternative emerge, Pivotal would be positioned as a *disruptor*, rather than as a *disruptee*. An aggressive shift to services on their own terms would place Pivotal well ahead of the curve. Other commercial vendors, forcibly jolted into the realm of services at a later date, would be hard pressed to compete.

12.3 Foster a "Community" of Pivotal Users

A middle-of-the-road option, hedged between the two extremes, would be for Pivotal to integrate aspects of the open source philosophy into its current operations. The aim would be to gain some of the benefits of open source without having to take any drastic measures. Encouraging user-driven modifications on a limited scale would foster a community of Pivotal users. Pivotal's role would then become that of an innovation arbitrator, deciding which changes should and should not be funnelled back into its commercial product.

The benefits of such an arrangement to customers are twofold. First, it offers customers greater customisation possibilities. Second, it prevents such user-driven customisations from being "orphaned" in future software releases. For Pivotal the benefits would be increased lock-in and reduced dependence on license-based revenues. There would also be another, less obvious advantage. Developing a community of Pivotal users would be an effective means of acquiring "sticky" information from customers, which would give Pivotal an important edge over commercial competitors.[21] Several risks would accompany such a move. Integrating user innovations calls for different internal skill sets and capabilities than the ones Pivotal's product development focus has cultivated. Doing so would therefore require the development of skills outside Pivotal's area of expertise. Moreover, there is a possibility that this option would not be going far enough to stave off the threat from source CRM, should such a threat materialise.

12.4 Emphasize Non-License Based Revenue Streams

Pivotal's financial statements and sales data point to a pattern, reflected in the industry as a whole. There has been a steady decrease in license revenues relative to total revenues over the last four years (AMR Research, 2004). Furthermore, our conclusions cast doubt on whether license-based revenue streams are sustainable. They are likely to

[21] This refers to tacit information that is difficult to transmit from users to developers – unless users become developers.

dry up with the dawn of open source CRM. Even without changing their proprietary software model, Pivotal would still need to address this trend.

There are several possible revenue streams aside from those based on license fees – most of which Pivotal already pursue to some extent. They include charging maintenance and support fees, offering premium customisation services, providing training and education services, and even consulting. This option involves placing more emphasis on these service-based revenue streams, with the objective of reducing dependence on license fees. Doing so would make Pivotal less vulnerable to expected future downswings in license revenues. Additionally, this approach would cultivate expertise in service areas – important since, in an open source driven environment, competition will revolve around services. A shortcoming of this approach is that it extrapolates based on recent trends to conclude that license revenues will decrease indefinitely. It may be that after a short-term drop, license revenues will rise in importance. If that is the case, this option would be steering Pivotal in the wrong direction.

12.5 Support Open Source Side Project

A derivative of option 2 is for Pivotal to quietly support its own open source project. This side project could feature a watered-down version of Pivotal's commercial product, or perhaps an older, less sophisticated release of it. The purpose would be to test the waters. Having an open source side project would provide valuable experience and develop different capabilities – both of which would leave Pivotal well positioned should open source CRM succeed in disrupting the commercial CRM market.

Keeping it separate from core business activities would be important for two reasons. First, in order to isolate it from the company's commercial operations. Otherwise, Pivotal's proprietary software mentality could prevent it from achieving its potential as something deliberately different. Second, to prevent against its cannibalising Pivotal's commercial business. Even with such separation, however, there is a risk that having a Pivotal open source project available – albeit a less capable one – could negatively impact commercial interests. Additionally, it could be argued that any resources allocated to this side project would be resources diverted away from Pivotal's core business, resulting in a less efficient, less focused company strategy.

12.6 Create Scaled-Down Product Line

A variation of the previous option would be to create a new scaled-down Pivotal product in order to appeal to the CRM non-user segment. The difference is that instead of it being an open source side project, Pivotal could keep it in-house and do so without expressly emphasizing an open source element to it. The new "Pivotal-lite" product would, however, have to be offered at an affordable price in order to achieve its purpose of appealing to non-users. It could even, perhaps, be offered at no charge, with its revenue coming exclusively from accompanying services, such as implementation and maintenance. Additionally, a scaled-down offering could serve as an intermediary stepping stone. Non-users, who may not otherwise consider purchasing commercial CRM software from Pivotal or anyone else, may be persuaded to at least try a free Pivotal-lite product. After seeing the benefits of CRM first-hand, some customers may then decide to upgrade to Pivotal's more sophisticated commercial offering.

The same drawbacks highlighted with regards to the previous option apply here as well. Resources are not unlimited. Developing and supporting this scaled-down product would be tying up resources that could have otherwise been spent advancing Pivotal's commercial products. Moreover, openly offering a free or inexpensive software alternative may anger Pivotal's higher-priority, paying customers.

13 EVALUATION & RECOMMENDATIONS

We presented six viable options that Pivotal could follow to address the threat posed by open source CRM. Each option has its advantages and disadvantages. In order to evaluate them, they need to be measured against Pivotal's overriding objectives.

13.1 Evaluation Criteria

Pivotal's objectives are profitability and longevity – in other words, sustainable profitability. This is still too vague for our purposes. What does Pivotal need to do in order to achieve sustainable profitability in light of our analyses? Right now, Pivotal can use its performance advantages to effectively compete with and outdo open source CRM in most market segments. If open source CRM manages to narrow the performance gap, as our analyses have indicated it will, Pivotal will no longer be able to rely on the crutch of better performance. The company will need to be able to compete along different dimensions as well. Two long-term objectives are: (1) increasing its appeal to non-users; and (2) achieving greater strategic flexibility. Gaining new customers relates to profitability. Increasingly, the battleground for new customers will be fought over the CRM non-user segment. Strengthening its appeal to non-users will therefore be necessary in order to compete with open source CRM. To sustain profitability over the long-term, Pivotal will need to be flexible. Our analyses suggest that the industry will undergo dramatic changes. As such, being able to adapt to changing circumstances will be necessary in order to sustain profitability. Drilling down further, these high-level objectives map to the following goals:

- **Increase customisation options:** One of the CRM industry's fundamental problems identified in *Section 8* is the lack of customisation options available. It is also one of open source CRM's main advantages over commercial products. Pivotal will thus need to find ways of offering customers greater customisation in order to compete with an open source CRM alternative.

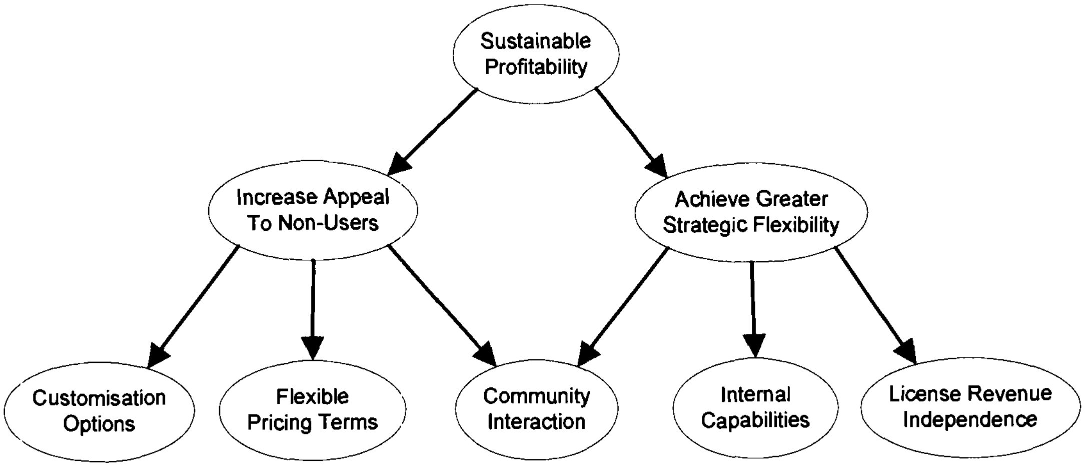
- **Offer flexible pricing:** A second fundamental problem was the unfavourable payment schedule from the customer's viewpoint. For Pivotal to widen its appeal to non-users, it has to be able to offer greater flexibility in pricing arrangements with customers.

- **Build up community interaction:** A critical advantage of open source software is its ability to transfer "sticky" information from the minds of customers to future product releases. Harnessing community interaction will be a major competitive requirement as the CRM industry moves forward.

- **Develop internal capabilities:** As part of building greater strategic flexibility, Pivotal needs to develop the internal skills and capabilities necessary to be able to change directions as circumstances dictate.

- **Decrease dependence on license revenues:** Another key element in achieving strategic flexibility is reducing dependence on license revenues. With the relative importance of license revenues expected to decrease for the foreseeable future, firms that depend on license revenues to support large marketing and sales departments will be limited in their abilities to react to a changing competitive environment.

Figure 13.1 Pivotal's Hierarchy of Goals and Objectives


Sustainable Profitability
Increase Appeal To Non-Users
Achieve Greater Strategic Flexibility
Customisation Options
Flexible Pricing Terms
Community Interaction
Internal Capabilities
License Revenue Independence

13.2 Matrix of Alternatives

These goals provide the criteria along which each strategic option can be evaluated. Two additional factors to consider are each option's feasibility and risk level. Options that involve drastic changes to Pivotal's strategy are less feasible. While still potentially viable, they entail greater costs. The "feasibility" and "risk level" criteria are intended to capture such costs.

Table 13.1 Matrix of Alternatives

Options	Objectives					Costs	
	Custom. Options	Community Interaction	Flexible Pricing	Internal Capabilities	Revenue Streams	Feasibility	Risk Level
Maintain Status Quo	✓	✓			✓		××
Adopt a Full Open Source Strategy	✓	✓✓✓	✓✓	✓✓✓	✓✓	×××	×××
Foster Community of Pivotal Users	✓	✓✓		✓✓		×	×
Find & Develop Additional Revenue Streams	✓		✓	✓	✓✓✓		×
Support Open Source Side Project	✓✓	✓✓	✓✓	✓✓	✓	××	××
Create Scaled-Down Product Line		✓	✓✓		✓		×

13.3 Discussion

The first option – maintaining the status quo – does not do much to further the five goals identified. It does, however, represent the safest option. No major changes would need to be made; no one would have to be convinced about having to make major changes. Such a course of action (or rather inaction) does carry risks, though. A competitive position that is acceptable today may be unacceptable tomorrow – a phenomenon referred to as a vanishing status quo.[22] If Pivotal does nothing to prepare

[22] Also sometimes referred to as "The Red Queen Effect" based on the line from Alice in Wonderland, "You have to run faster and faster just to stay in the same place!" (Kauffman, 1995)

itself for the emergence of a competitive open source alternative, the company will be poorly equipped to respond when it can no longer avoid doing so.

The second option of adopting a full open source strategy goes furthest in preparing Pivotal for a changed future landscape. Not surprisingly, it also carries the heaviest costs. First, in terms of the feasibility of actually implementing it; second, it would require core skills and capabilities foreign to Pivotal's current product-oriented operations. Given the risks involved, hard evidence would be needed to justify such a drastic step. It would therefore be best to table this option for the time being. It could be reconsidered later as circumstances change.

The intermediary third option splits the difference. As a preparatory step, it would serve to build the internal skills necessary to compete with an open source alternative. The risks associated with this option are less severe, since it would not require a change from Pivotal's proprietary software model.

Emphasizing additional revenue streams would reduce Pivotal's dependence on license fees. Doing so would have the added benefits of building internal skills around services, enabling more flexible pricing alternatives, and increasing customisation possibilities. Given that Pivotal already provides non-license based services, this course of action would be an extension of current strategy, which makes it highly feasible. There are risks associated with it, however. In the event that the recent downward trend on license revenues changes, shifting focus towards non-license based revenue streams would be a strategic misstep.

The fifth option – supporting an open source side project – would enable Pivotal to gain the benefits of open source and develop internal expertise, without committing the company's future towards it. Having a stake in both commercial and open source CRM would leave Pivotal in a good position to capitalise – whether open source CRM emerges as a viable alternative or not. Doing so, however, may involve stretching Pivotal's resources too thin. Maintaining a full-blown commercial product and supporting a separate open source product would involve a juggling act that would not be sustainable for very long. There is also the risk of alienating both employees and customers.

The final option of creating a scaled-down product line would offer some of the benefits of the previous option, but with costs that are less severe. Pivotal's appeal towards the non-user segment would be strengthened via its more flexible pricing alternatives. On its own, this option does little to build internal capabilities or to foster community interaction, though. Still, costs are minimised for this very reason. It does not represent a major departure from Pivotal's current strategy, but it would enable the company to compete more effectively over the non-user segment.

13.4 Recommendations

Note that some but not all of the options are mutually exclusive. Maintaining the status quo and adopting a full open source strategy clearly are, but some of the others can be pursued in conjunction. Options 3, 4, and 6, in particular, represent relatively low risk courses of action that, taken together, would address all five strategic goals.

We recommend that Pivotal: (1) foster a community of Pivotal users; (2) develop non-license-based revenue streams; and (3) create a scaled-down product line. These three recommendations would strengthen Pivotal's position in anticipation of an emerging open source CRM alternative, the decreasing significance of license revenues, and the growing importance of the non-user segment.

14 MAKING A BUSINESS CASE FOR CHANGE

It is difficult to advocate a strategic change of direction based on an as yet unrealised threat from open source CRM. At present, open source CRM alternatives, such as SugarCRM, are not sophisticated enough to represent a genuine alternative for Pivotal's customers. Open source CRM simply does not meet their more exacting requirements.

By examining four open source case studies, however, we have seen that it is possible for open source projects to enter markets as sub par performing alternatives and still succeed. They gained initial user bases by appealing to a market segment neglected by incumbents: non-users. For such price sensitive users, the adoption choice was not between open source and superior performing commercial competitors – it was between open source and nothing. Although these initial users did not provide the four open source projects with revenue, they did provide valuable experience. They tested the products, provided feedback, and made modifications, which were then funnelled back into the community, enabling the open source projects to rapidly narrow their performance gaps with commercial incumbents. Progressively more demanding market segments came to see open source as a viable alternative. Before long, commercial incumbents that had initially dismissed the threat from open source were losing market share. With the performance gaps narrowed, they were unable to rely on superior performance to shield themselves from competition from more economical open source alternatives.

Similarly, there is a group of potential CRM users with lower performance expectations than what commercial incumbents are offering – very small companies and non-profit organisations, for example. They cannot afford to pay high price tags for CRM software and, as a result, they are neglected by commercial incumbents. For them, likewise, the adoption decision is not between open source CRM and Pivotal, or SAP, or Oracle – it is between open source CRM and nothing. That commercial vendors offer better performance is not a consideration; price trumps performance for this group of users.

Open source CRM is already gaining traction with non-users. It is quite possible that it will follow a similar disruptive path as the four case studies outlined above. There are three fundamental problems with the CRM industry as it is now, from customers' perspective, which could provide impetus towards such a shift towards open source: (1) extreme vendor lock-in; (2) unfavourable payment schedules; and (3) a lack of customisation options. Moreover, the events that could lead to a narrowing of the performance gap are all very plausible. Given their interrelated and self-reinforcing nature, a narrowing of the performance gap is likely.

In light of these conclusions, treating open source as a legitimate competitor and threat is worth the risk; the potential costs of inaction are simply too high. We recommend that Pivotal:

(1) Foster a community of Pivotal users

(2) Develop non-license-based revenue streams

(3) Create a scaled-down product line

Fostering a community of Pivotal users will serve to improve two-way interaction with customers, to develop Pivotal's capabilities as an innovation arbitrator, and to transfer "sticky" customer information into future product releases. Developing alternative revenue streams will reduce Pivotal's dependence on license fees. Doing so will have the salutary effect of increasing future strategic flexibility. Creating a scaled-down product line will enable Pivotal to appeal to the non-user segment that is currently beyond reach due to financial considerations. These three recommendations are each realistic and feasible from Pivotal's current strategic disposition. Together, they form a course of action that is both bold and aggressive, but without being overly drastic or impractical.

APPENDICES

Appendix A: CRM Software Total Cost of Ownership

Figure A.2 CRM Software's First Year Total Cost Comparison

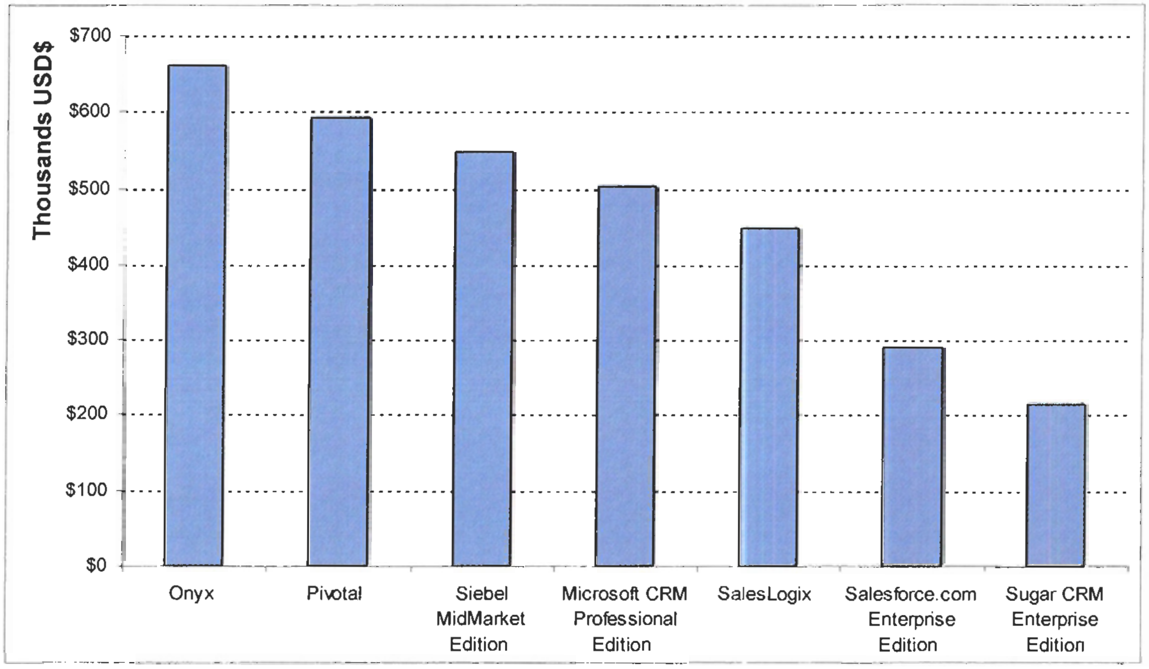
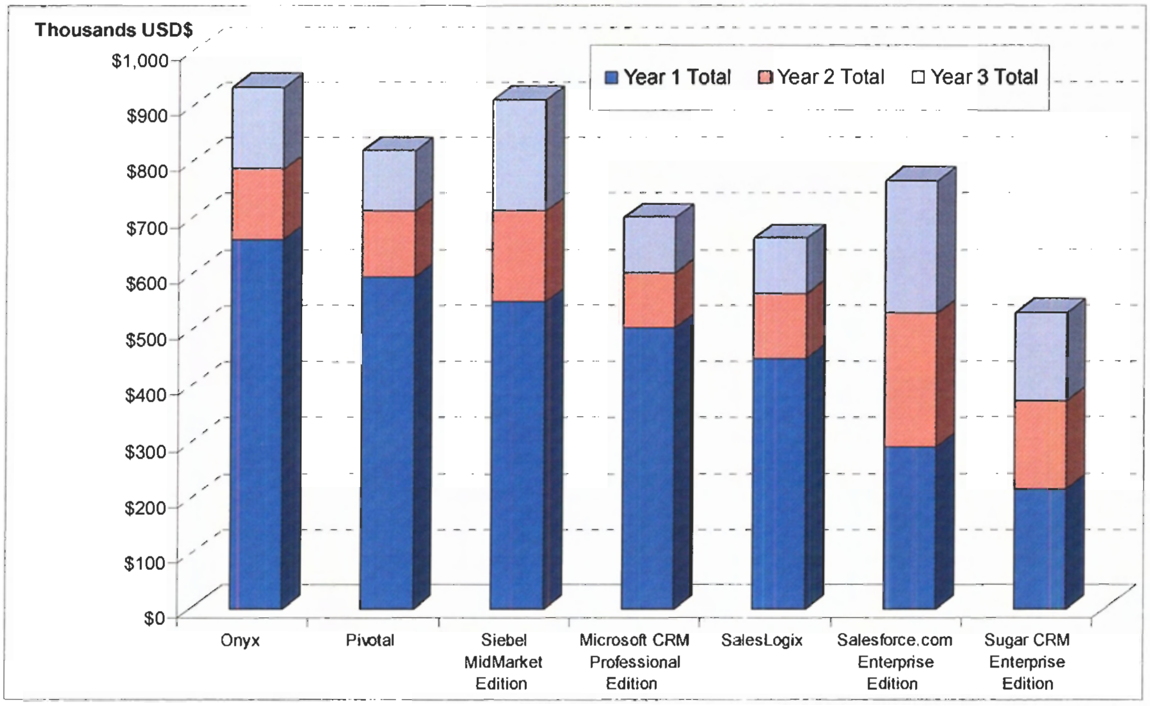


Figure A.3 CRM Software's 3-Year Total Cost of Ownership Comparison



Based on Gartner Three-Year TCO Scenario & data publicly available at www.sugarcrm.com

Appendix B: Revenue Figures for Pivotal & Salesforce.com

Table A.1 Pivotal's Quarterly Revenues

Quarter	Quarterly Revenue
Mar-98	$ 3,930,000
Jun-98	$ 5,128,000
Sep-98	$ 4,568,000
Dec-98	$ 5,713,000
Mar-99	$ 6,723,000
Jun-99	$ 8,323,000
Sep-99	$ 8,675,000
Dec-99	$ 11,542,000
Mar-00	$ 14,537,000
Jun-00	$ 18,185,000
Sep-00	$ 21,220,000
Dec-00	$ 26,051,000
Mar-01	$ 26,639,000
Jun-01	$ 22,244,000
Sep-01	$ 16,130,000
Dec-01	$ 16,698,000
Mar-02	$ 17,716,000
Jun-02	$ 19,072,000
Sep-02	$ 12,307,000
Dec-02	$ 16,095,000
Mar-03	$ 13,135,000
Jun-03	$ 14,540,000
Sep-03	$ 13,302,000
Dec-03	$ 14,135,000

Based on Pivotal SEC filings prior to CDC Acquisition

Table A.2 Salesforce.com's Annual Growth and Revenues

Year	Subscription and support	Professional Services and other	Total Revenue	% Change
2001	$ 5,022,000	$ 413,000	$ 5,435,000	-
2002	$ 21,513,000	$ 896,000	$ 22,409,000	312%
2003	$ 47,656,000	$ 3,335,000	$ 50,991,000	128%
2004	$ 85,796,000	$ 10,227,000	$ 96,023,000	88%
2005	$ 157,977,000	$ 18,398,000	$ 176,375,000	84%

Based on Salesforce.com SEC filings

REFERENCE LIST

Agrawal, A. & Garlappi, L. (2002). "Public sector science and the strategy of the commons." *Academy of Management Proceedings.* BPS X1-6.

AMR Research Customer Management Application Team (2004). "The customer management applications report, 2003–2008." *AMR Research Inc.,* AMR-R-17328.

Augustine, L. (2005). "The next wave of open source: Applications." *IT Conversations,* recorded April 5. Retrieved September 15, 2005 from http://www.itconversations.com/shows/detail493.html.

Band, W., Kinikin, E., Ragsdale, J., & Harrington, J. (2005). "Enterprise CRM suites, Q2, 2005: evaluation of top enterprise CRM software vendors across 177 criteria." *Forrester Research Inc.,* July 13, 2005.

Christensen, C. M. (1997). *The innovator's dilemma: When new technologies cause great firms to fail.* New York: Harvard Business School Press.

Christensen, C. M. & Raynor, M. E. (2003). *The innovator's solution: Creating and sustaining successful growth.* Boston: Harvard Business School Press.

Christensen, C. M., Anthony, S. D. & Roth, E. A. (2004). *Seeing what's next: Using the theories of innovation to predict industry change.* Boston: Harvard Business School Press.

Close, W. (2003). "CRM suites for North American MSBs markets: 1H03 magic quadrant." *Gartner Inc. Markets,* M-19-3669, April 3.

Close, W., Eisenfeld, B., Davies, J., & Bona, A. (2004). "The three-year total cost of ownership for CRM software for MSBs." *Gartner Inc. Strategic Analysis Report,* R-22-6895, April 15.

Cyberspace (2004). "Linux versus Windows: Total cost of ownership comparison." Retrieved November 14, 2005, from http://members.iinet.net.au/~cybersrc/about/linux_vs_windows_tco_comparison.pdf.

Fano, R. M. (1967). "The computer utility and the community." *IEEE International Convention Record,* 30-34. Excerpts reprinted in IEEE Annals of the History of Computing, 14 (2), pp. 39-41. Piscataway, NJ: IEEE Educational Activities Department.

Hippel, E. v. & Krogh, G. v. (2003). "Open source software and the 'private-collective' innovation model: Issues for organization science." *Organization Science,* 14 (2), pp. 209-223.

Hippel, E. v. (2005). *Democratizing Innovation.* Cambridge, MA: MIT Press.

IDC (2002). "Windows 2000 versus Linux in enterprise computing: An assessment of business value for selected workloads." *IDC White Paper.* Retrieved October 29, 2005 from http://download.microsoft.com/download/d/b/8/db8543a5-1e19-42e6-b0e3-d17ae2e2a9d2/IDC20TCO20Paper.pdf.

JavaWorld.com (2002). *Will Big Blue eclipse the Java tools market?* Retrieved October 20, 2005 from http://www.javaworld.com/javaworld/jw-02-2002/jw-0222-eclipse.html.

Kauffman, S. A. (1995). "Escaping the red queen effect." *The McKinsey Quarterly,* 1, pp. 119-129.

Lerner, J., & Tirole, J. (2000). "The simple economics of open source." *National Bureau of Economic Research,* NBER Working Paper No. 7600.

Oster, Sharon M. (1999). *Modern Competitive Analysis.* (3rd ed.). New York: Oxford University Press.

Raymond, Eric S. (2000). *The Cathedral and the Bazaar.* Retrieved September 28, 2005 from http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html.

Relevantive (2003). "Linux usability study report." *Relevantive AG 2003.* Retrieved November 01, 2005, from http://www.linux-usability.de/download/summary_linux_usability.pdf.

Tuomi, I. (2005). "The future of open source." In Wynants, M. & J. Cornelis (Eds.), *How Open is the Future?* (pp. 429-459). Brussels University Press.

Wardley, M., & Bulmstein, R. (2004). "Worldwide CRM applications 2004-2008 forecast update and 2003 vendor shares." *IDC Software Research Group: Market Analysis*, IDC #31524 (1).