

# The Emerging Market for Web-based Enterprise Software

by

**Andreas Göldi**

lic. oec. HSG

University of St.Gallen, Switzerland

Submitted to the MIT Sloan Fellows Program in Innovation and Global Leadership  
in Partial Fulfillment of the Requirements for the Degree of

**Master of Science in Management of Technology**

At the

**Massachusetts Institute of Technology**

June 2007

©2007 Andreas Göldi. All rights reserved.

The author grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature of Author \_\_\_\_\_

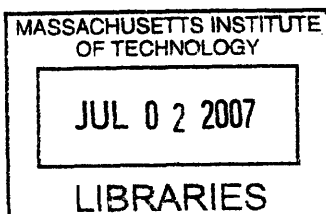
**Andreas Göldi**  
MIT Sloan School of Management  
May 11, 2007

Certified by \_\_\_\_\_

**Michael A. Cusumano**  
Thesis Supervisor  
Sloan Management Review Distinguished Professor of Management

Accepted by \_\_\_\_\_

**Stephen Sacca**  
Program Director  
MIT Sloan Fellows Program in Innovation and Global Leadership



**ARCHIVES**

# **The Emerging Market for Web-based Enterprise Software**

by

Andreas Göldi

Submitted to the MIT Sloan Fellows Program in Innovation and Global Leadership  
on May 11, 2007 in Partial Fulfillment of the Requirements  
for the Degree of Master of Science in Management of Technology

## **Abstract**

Web-based enterprise software – sometimes referred to as “Software as a Service” (SaaS) or “on demand software” – is a major wave of innovation that introduces a new technical and economic model to enterprise software. The defining characteristics of web-based enterprise software are: a fully web-based user interface, hosted application deployment, a SaaS-based business model, and the use of a service-oriented architecture for integration.

This study analyzes a sample of 108 companies currently offering SaaS-based products. In total, 35 different application types and 20 different combinations of revenue models were counted. Apparently, the market is still in an experimental phase, and truly dominant designs haven’t emerged yet. Gross margins for SaaS-oriented companies are lower than for traditional software product companies, but still attractive at around 85%.

The entrepreneurial activity in the sector is significant. More than 50% of the companies in the sample were founded in or after 2002, and almost 30% in the last two years. Venture capital continues to be an important source of capital with 36% of the companies having received VC investment, but an equally large percentage of companies are bootstrapped, i.e. have no formal source of outside capital.

The data about customer adoption of web-based enterprise software is not very clear. Most studies suggest that SaaS accounted for about 5% of the CRM market in 2006, with other application types below that level. In total, SaaS probably doesn’t even account for 1% of the global software market. However, customer willingness to adopt SaaS is apparently rising very quickly, and specialized SaaS companies are experiencing rapid growth.

From the data available, it can’t be decided unambiguously if web-based enterprise software is a truly disruptive model or merely an incremental innovation. Most characteristics point to a new-market disruption, i.e. an innovation that will bring new functionality to current non-users of advanced enterprise software.

Thesis Advisor: Michael A. Cusumano

Title: Sloan Management Review Distinguished Professor of Management

## **Acknowledgments**

First and foremost, I would like to thank my thesis advisor Prof. Michael A. Cusumano. His work has always been a major influence throughout my career, and I consider myself honored to be his student. His class “The Software Business” at MIT Sloan gave me the inspiration for my thesis project. Prof. Cusumano supported me with many insights and suggestions throughout the work of this thesis.

I also would like to thank my classmates at MIT Sloan for the inspiring work we did together, particularly Bettina Hein, Tim Jones, Bobby Lo, Paul Hsu and Caleb Hug. The team projects we worked on collectively provided many of the initial ideas for this thesis. Furthermore, I would like to thank Prof. Erik Brynjolfsson, whose class “The Economics of Information” gave me the opportunity to work on a student project for Google that provided many valuable insights.

Finally, I would like to thank the MIT Sloan Fellows Program in Innovation and Global Leadership. This year has been one of the best of my life, and I am deeply grateful for this opportunity.

This thesis is dedicated to my wife Bettina who has always supported me in my professional, academic and private activities. Her energy, perseverance and love are my most important inspiration.

## Table of Contents

1	Introduction.....	8
2	Research Question and Methodology.....	10
3	Traditional Enterprise Software.....	11
3.1	Definition: Enterprise Software.....	11
3.2	The Market for Enterprise Software.....	12
3.2.1	Short History.....	12
3.2.2	Application Types.....	12
3.2.3	Main Players in the Enterprise Software Market.....	15
3.2.4	Market Development.....	17
4	Web-based Enterprise Software.....	19
4.1	Historical Background.....	19
4.2	Web 2.0 and Enterprise Software.....	22
4.3	Characteristics of Web-based Enterprise Software.....	23
4.3.1	Web-based User Interface.....	23
4.3.2	Hosted Application Deployment.....	24
4.3.3	“Software as a Service” Business Model.....	27
4.3.4	Service-oriented Architecture for Integration.....	29
4.4	Summary.....	31
5	Market Structure.....	32
5.1	Sample Description.....	32
5.2	Framework for the Classification of Web-based Enterprise Applications.....	33
5.2.1	Application type.....	33
5.2.2	Revenue model.....	37
5.2.3	Server Deployment Model.....	38
5.2.4	Client Deployment Model.....	38
5.2.5	Integration Model.....	39
5.2.6	Vendor Attributes.....	39
5.3	Vendor Market.....	40
5.3.1	Sample Analysis.....	40
5.3.2	Conclusions from Sample Analysis.....	51
5.4	Case Studies.....	51
5.4.1	Salesforce.com.....	52



5.4.2	RightNow Technologies.....	56
5.4.3	NetSuite.....	58
5.4.4	Xactly .....	59
5.4.5	Smartsheet.com.....	59
5.4.6	37signals.....	60
5.4.7	Zoho / AdventNet .....	61
5.4.8	Microsoft Office Live .....	62
5.4.9	Google Apps.....	63
5.4.10	SAP .....	64
5.4.11	Oracle .....	65
5.4.12	Summary of Case Studies.....	65
5.5	Customer Adoption of Web-Based Enterprise Software.....	69
5.5.1	Current Adoption and Market Size.....	69
5.5.2	Predictions.....	71
5.6	Disruptive Characteristics .....	71
5.7	Emerging Dominant Designs .....	74
6	Strategic Framework .....	76
6.1	The Vendor Perspective.....	76
6.1.1	Customer and Market Strategy .....	76
6.1.2	Business Model Strategy.....	79
6.1.3	Technology Strategy .....	83
6.1.4	Summary .....	84
6.2	The Customer Perspective .....	85
6.2.1	Application Type .....	86
6.2.2	Deployment Scope.....	86
6.2.3	Vendor Type.....	86
6.2.4	Customization and Integration.....	87
6.2.5	Contractual Aspects and Billing Model .....	87
6.2.6	Service and Support Strategy .....	87
6.2.7	Risk Management .....	88
7	Summary and Conclusion.....	89
8	Appendix.....	91
9	References.....	99

## Tables

Table 1: Main characteristics of network-centric application generations.....	21
Table 2: Application types for horizontal enterprise processes.....	34
Table 3: Application types for personal productivity .....	35
Table 4: Application types for communication and collaboration.....	36
Table 5: Application types for vertical and specialized applications.....	36
Table 6: Application types for infrastructure and application development.....	36
Table 7: Revenue models .....	37
Table 8: Server deployment models.....	38
Table 9: Client deployment models .....	38
Table 10: Integration models.....	39
Table 11: Vendor attributes .....	39
Table 12: Frequency of application types .....	41
Table 13: Revenue model combinations .....	44
Table 14: Summary of case studies.....	67
Table 15: Comparison of software costs for Salesforce.com and Siebel CRM.....	73

## Figures

Figure 1: ERP revenues 1995-2006, US \$ billion (AMR Research, 2006).....	18
Figure 2: Historical development of alternative paradigms for software deployment.....	19
Figure 3: Application type clusters .....	42
Figure 4: Revenue models .....	43
Figure 6: Deployment model combinations .....	46
Figure 7: Year of company formation.....	47
Figure 8: Primary source of financial capital .....	48
Figure 9: Location of headquarters .....	50
Figure 10: Salesforce.com user interface .....	52
Figure 12: Revenue mix for Salesforce.com and SAP AG .....	55
Figure 13: Cost structures of Salesforce.com and SAP AG.....	56
Figure 14: Income Statement for RightNow Technologies.....	57
Figure 16: 37signals Basecamp project management application.....	60
Figure 17: Zoho online word processor .....	61
Figure 18: Microsoft Office Live CRM application .....	63

Figure 19: Google's e-mail application with advertisements ..... 64  
Figure 20: Strategy pyramid for web-based enterprise software ..... 85

# 1 Introduction

*„Technology always develops from the primitive via the complicated to the simple.“*

*-- Antoine de Saint-Exupéry*

The software industry, over the course of its relatively brief history, has experienced several major disruptions. Almost every decade or so, a new base technology, business model or approach to deployment changes the rules of the game. This started back in the 1960s when software first was sold as a separate product from hardware (Cusumano, 2004). Since then, the software industry has experienced the rise of the mini computer and later the personal computer, the invention of client/server computing, the integrated enterprise software package, the World Wide Web, service-oriented architectures and many smaller innovations in between these major waves.

Probably the consistently most successful sector in the software industry is enterprise software. Companies in this sector sell their products to corporate customers rather than to individuals. Enterprise software products help to organize and improve a company's many processes: accounting and resource planning, HR, internal collaboration, office work, supply chain management, sales force management, and so on. Another important part of the enterprise software space are products that provide a stable and scalable technical base for these process-oriented software products: databases, middleware, workflow software, etc.

Interestingly, most dominant enterprise software product companies in today's market<sup>1</sup> have their origins in the 1970s: Microsoft, Oracle, SAP and Computer Associates were all founded 30 years or more ago; a few others, such as Symantec, Intuit and Adobe were founded in the early 1980s. Many of the stars of the "new economy" bubble of the late 1990s experienced rapid growth, but were particularly badly affected by the IT crisis early in the current decade. They had to scale down massively or were acquired by other companies. Well-known examples include Siebel Systems, i2 Technologies, or Ariba (Foremski, 2001).

Based on this rather static market situation, it could be argued that enterprise software is a mature market and that there haven't been any major disruptive waves since the 1980s that would have allowed new competitors to build large, sustainable businesses. The most successful companies of the Internet age – Google, Yahoo, eBay and Amazon.com come to mind – are typically not considered to be software companies. They make most of their money from advertising or e-commerce revenues.

However, Internet technology has enabled a new, currently emerging wave of web-based enterprise software products that has the potential to threaten the dominance of the existing players. This new breed of software is fully based on web technology, usually delivered as a hosted service over the Internet and typically charges customers a monthly or

---

<sup>1</sup> Selected from the Forbes Global 2000 list (Forbes, 2005)

annual subscription fee instead of up-front license payments. The advantages for customers are obvious: They save on server hardware, installation costs and up-front investments for software licenses and customizing. Although most of these new software products currently offer only limited functionality, their advantages in cost and simplicity could pose a significant challenge to established software companies.

Ray Ozzie, Microsoft's new Chief Software Architect, acknowledged this threat recently in an internal memo titled "The Internet Services Disruption" (Ozzie, 2005). Microsoft, Oracle, SAP and others are working on new product generations that work with Internet technology and subscription-based business models. On the other side, new companies such as Salesforce.com are showing promising early results (Salesforce.com, 2006), and some of the big Internet players, particularly Google, have started to offer first products in the enterprise software space (Google, 2007a).

This thesis will examine the emerging market for web-based enterprise software and try to identify characteristics of this new type of software. It will also try to provide some useful frameworks for research and strategy development in this field.

The thesis is divided into seven chapters. After this introductory chapter, the research question and methodology will be outlined in chapter 2. Chapter 3 will briefly describe the market for traditional enterprise software, since web-based applications are rooted in and compete against older types of business applications. Chapter 4 will examine the specific characteristics of web-based enterprise software.

In chapter 5, a sample of 108 vendors of web-based applications will be analyzed. The results will be used to establish a framework for the classification of these applications, to analyze the sample in detail and draw conclusions about the market structure and characteristics of this market. Chapter 6 will introduce a strategic framework for both vendors and buyers of web-based applications, and chapter 7 will summarize the thesis and draw some final conclusions.

## 2 Research Question and Methodology

The goal of this thesis is to explore the emergence of a new type of enterprise software. The key research question is: Is web-based enterprise software a major disruptive innovation that will change the structure of the enterprise software market, or is it merely an incremental innovation that can easily be adapted by established companies? And furthermore, how should software vendors and software buyers react to this new form of software?

The motivation to answer these questions is twofold: First of all, disruptive innovations have traditionally dramatically expanded the usefulness of a certain type of product by making these products available to a bigger group of customers for much lower costs (Christensen, 1997). A major shift in the enterprise software market could mean that the functionality of such software packages become affordable for smaller companies and cost-efficient for broader areas of application in bigger companies (Vincent et al., 2005). This would almost certainly have a productivity-boosting effect (Aral, Brynjolfsson and Van Alstyne, 2006).

Secondly, major waves of innovation have always created successful, very rapidly growing new companies, resulting in massive wealth creation. Prime examples for this effect are Microsoft, Google, or Oracle. This type of dominant company not only generates wealth directly for its shareholders, but also causes the growth of a whole ecosystem of complementor companies, thereby creating whole sub-industries (Gawer and Cusumano, 2002). These effects are obviously very beneficial for the overall economy. This thesis therefore aims to examine the characteristics and stage of the assumed entrepreneurial opportunity enabled by web-based software.

The research methodology used is built on two pillars:

- Current literature on the subject and other relevant public information (such as financial information from public software companies). These sources are the basis for chapters 3, 4 and parts of chapters 5 and 6.
- A sample of 149 software companies who offer web-based enterprise applications, collected from a number of public sources (see section 5.1). Based on previously defined criteria, this sample was reduced to 108 relevant companies that were further analyzed, and 11 companies were examined in detail using short case studies. These analyses are the basis for chapter 5 and parts of chapter 6.

## 3 Traditional Enterprise Software

### 3.1 Definition: Enterprise Software

The notion of enterprise software doesn't have a clear definition that is shared by many authors. The only common denominator is the fact that enterprise software is sold to companies, not to individuals (Cusumano, 2004).

Some authors, such as Craig Le Clair, use a definition that focuses on core processes in a company:

“Enterprise-level information systems are the information technologies (IT) that firms use to support their core business processes, such as sales, finance and operations.” (Le Clair, 2005)

This definition would be too restrictive for the subject at hand, since many web-based enterprise software products target processes that are not necessarily at the core of a company's business, but still very relevant for overall productivity.

A bit broader is Andrew McAfee's definition:

“Enterprise IT (EIT) is the type of IT application that companies adopt to restructure interactions among groups of employees or with business partners. Applications that define entire business processes, such as CRM and SCM—as well as technologies, such as electronic data interchange, that automate communications between companies—fall into this category.” (McAfee, 2006)

However, this definition still excludes the less structured processes, such as team collaboration (e.g. project management) and semi-structured communication (such as e-mail or knowledge management systems) that impact an increasingly important part of a company's value creation.

For the purposes of this thesis, I will therefore use the following definition:

*Enterprise software provides functionality that supports and enables core and support processes in an organization, such as planning, financial management, HR management, team collaboration, or customer relationship management.*

*Web-based enterprise software uses standardized, platform-independent web browsers as the front-end infrastructure and is typically (but not exclusively) deployed as a remotely hosted service.*

This definition implies that enterprise software can be used by organizations of any size. Since web-based enterprise software is often targeted to small and medium-sized businesses, this thesis will explicitly look at solutions for companies of every size. Furthermore, an enterprise software product doesn't necessarily have to cover processes that are truly enterprise-wide. Many successful software products started out as solutions for departments or workgroups, and this is also true for many new web-based products. Therefore, solutions with this more limited scope will also be discussed here.

## **3.2 The Market for Enterprise Software**

### **3.2.1 Short History**

The term “software” started to become popular in the late 1950s (Cusumano, 2004). Before that, computer programs were rarely thought of as being a product that had a value separate from the hardware it ran on. Also in the 1950s, the first U.S. software companies were founded, often as government contractors. In this phase, most of these companies sold custom programming services rather than software products (Campbell-Kelly, 2004).

Already in the 1960s, so-called time-sharing services offered computing capacity on a pay-per use basis. Although many of these services merely sold computer time, some offered simple pre-written application programs to their customers (Campbell-Kelly, 2004). In some ways, these services were early predecessors of modern web-based applications.

Probably the first real software product was “Autoflow”, a flowcharting program developed in 1964 by Applied Data Research (Campbell-Kelly, 2004). The first “horizontal” software product that accounted for more than \$100m in sales was the database management package Mark IV, developed in 1967 (Cusumano, 2004). However, it took this product a full 16 years to reach the \$100m sales mark. When IBM introduced its System/360 computer family in 1964 and four years later started unbundling software from hardware, the era of productized software had definitely arrived.

By 1980, there were already 6,104 vendors selling packaged software in the U.S., with total revenues of \$2.5 billion. The majority of the products, over 75%, were applications with a business focus (Campbell-Kelly, 2004). The software industry in other countries, particularly in the UK, France, Germany and Japan, experienced similar growth, although with a different focus (Cusumano, 2004).

The emergence of the personal computer brought along a new wave of explosive growth in the software industry. The market for PC software grew from \$140 million in 1981 to \$1.6 billion in 1984 (Cusumano, 2004). New application categories like spreadsheets were invented, and software became a mass-market product. The 1990s brought the rise of client/server-computing (Messerschmitt, 1999), which is still today the dominant technical model for enterprise applications.

The group of software companies that dominates today’s enterprise software market was for the most part founded in the second half of the 1970s or early 1980s (see profiles in section 3.2.3). The origins of these companies are varied: Some filled niches in IBM’s large ecosystem, others were pioneers of the PC revolution, and yet others started as services companies and only later started focusing on software products.

Extensive accounts of the history of software can be found in Michael Cusumano’s “The Business of Software” (Cusumano, 2004) and Martin Campbell-Kelly’s “From Airline Reservations to Sonic the Hedgehog” (Campbell-Kelly, 2004).

### **3.2.2 Application Types**

Based on the definition of enterprise software in section 3.1, this section will explore in more detail what types of applications enterprise software vendors sell.



### **3.2.2.1 Horizontal Enterprise Processes**

The most common type of enterprise applications are software products that support or enable a horizontal, enterprise-wide business process, i.e. a process that is not specific to a particular type of business, but can be found in almost any kind of company and is used in more than one department or function.

Typical examples are:

- Accounting and financial planning
- Customer relationship management
- Supply chain management
- Inventory management
- Business process management
- Human resources management
- Business intelligence (reporting and data analysis)

Integrated product suites that integrate several of these functions into a common platform are called “Enterprise Resource Planning” (ERP) systems (Davenport, 1998).

### **3.2.2.2 Vertical and Specialized Applications**

In addition to horizontal applications, companies frequently use specialized software that supports a particular, focused business function that is found in several industries. Examples include (McAfee, 2006):

- Computer-aided design (CAD) software
- Simulation software, e.g. finite elements simulations
- Desktop publishing software

In many industries, companies use industry-specific “vertical” enterprise software packages that support particular core processes that are found only in that industry.

Typical examples:

- Patient record management for hospitals
- Insurance management software
- Professional service automation
- Hotel management software

### **3.2.2.3 Communication and Collaboration**

The first two categories of enterprise software are used to enhance well-defined, mostly structured processes. However, many of the activities in a company are unstructured and not easy to plan. Most of these work processes revolve around the communication between individuals or teams, the collaboration inside a department or company, or collaboration across organizational boundaries.

Software that enables these communication and collaboration processes is getting more and more important and complex. This segment of the software market started with basic e-mail and project management software, but now encompasses a whole range of product types.

Examples:

- E-Mail software
- Group calendaring software
- Instant messaging and conferencing software
- Project management applications
- Document sharing / content management
- Knowledge management

#### **3.2.2.4 Personal Productivity**

Another important type of software product is used primarily by individual users to enhance personal productivity. This type of software is sometimes not included in the definition of enterprise software. But since these applications are increasingly getting integrated with “classical” enterprise software, they should be part of the analysis framework. Furthermore, individually used applications have historically been the first step for the development of enterprise-wide application types. For example, calendaring software for individual purposes over time developed into company-wide calendaring solutions.

Examples:

- Word processing
- Spreadsheets
- Presentation software
- Calendars
- Task-planning software
- E-Learning software

#### **3.2.2.5 Infrastructure and Application Development**

Finally, the tools that are used to create and run applications define a secondary segment of the enterprise software market. This type of product will not be discussed in depth in this thesis, but since many of the leading companies in the enterprise software market offer development tools and even infrastructure elements, it is still important to mention this sector.

It could even be argued that the very important building of “ecosystems” around a major enterprise software product is only possible if a vendor offers adequate development tools to partners and third-party software companies. For example, SAP has traditionally offered its own software development environment to consulting firms and other software companies (SAP, 2007). This enabled other firms to customize and extend SAP’s solutions, thereby creating a more useful product.

Examples:

- Development tools
- Middleware
- Run-time environments for third-party modules
- APIs for third-party extensions

### **3.2.3 Main Players in the Enterprise Software Market**

In most cases, it is not a single company that is responsible for the complete development and deployment of an enterprise software solution. Much more typically, a base product is developed and marketed by a software firm, and different service-oriented companies install, customize and integrate the product for customers. Many enterprise software companies have a whole “ecosystem” of consulting partners that are trained to deploy the software firm’s products. For example, SAP currently lists 182 service partners in the United States alone (SAP, 2007).

This analysis will focus on companies that develop and sell the base products. This type of company is often referred to as a “product company”, as opposed to “service companies” that generate most of their revenues from services. However, most enterprise software companies now actually follow a mixed model. While they still derive most of their revenue and margin from product sales, they have a growing share of service revenues (Cusumano, 2004).

The following list of the top enterprise software companies is based on the Forbes Global 2000 list (Forbes magazine, 2005), from which the top software product companies were selected that have more than \$1 billion in revenue, 60% or more of which come from software license sales, including maintenance fees (based on these companies’ most recent 10-K annual report). The list focuses on vendors of business-oriented solutions. Companies that focus on infrastructure software for systems management and IT security (e.g. CA, BMC, Compuware, Symantec) were left out from this list.

#### **3.2.3.1 Microsoft**

Microsoft was founded in 1975 by Bill Gates and Paul Allen (Microsoft, 2007). The company first focused on development tools for the emerging personal computer market, particularly its BASIC programming language. Famously, the breakthrough for the young company came when IBM asked it to develop an operating system for the IBM PC.

Today, Microsoft is the dominant player in several segments of the software market:

- It is by far the dominant vendor of desktop operating system with a market share of well over 90% for its Windows OS (Thurrott, 2003).
- The Microsoft Office suite of personal productivity software is the de-facto standard for word processing, spreadsheets and presentation software (Delaney and Guth, 2006).
- Microsoft’s server products are also among the leading contenders in several markets. Its Windows NT family is the market leader for file and print servers (Shankland, 2006). The Microsoft Exchange e-mail and calendaring server soft-

ware has a global market share of approximately 30%, which gives it a No. 1 position (Radicati, 2006). Microsoft also has popular products in the markets for relational database software (SQL server), Web servers (Microsoft Internet Information Server), and system management.

- In the market for ERP software, Microsoft acquired several independent companies (Microsoft, 2007) and integrated their products into the Microsoft Dynamics product line. This family of applications covers CRM, supply chain management and financial management. Although Microsoft is among the fastest growing ERP vendors, its market share is still below 5% (AMR Research, 2006).

Microsoft's total revenues in 2006 were \$44.28 billion (Microsoft, 2007).

### **3.2.3.2 IBM**

IBM is a special case: Although the company generates less than 20% of its total \$ 91.4 billion (2006) in revenues from software sales, it is still the second-largest software company in the world with software revenues of \$18.16 billion (IBM, 2006).

IBM's software palette is strongly focused on infrastructure software such as application servers, middleware platforms, operating systems, and security software. Equally strong is IBM's position in application development and systems management software. But the company is also the second-largest player in e-mail and collaboration software with its Lotus Notes product family (Radicati, 2006).

IBM currently doesn't have any ERP offerings or other business-oriented applications in its software portfolio.

### **3.2.3.3 Oracle**

Oracle is best known as the market leader in relational database management (RDBMS) software (Bank, 2004). But over the last few years, the company has also built a strong position in business applications, mostly from major acquisitions of its rivals PeopleSoft and Siebel. Furthermore, the company has a broad palette of middleware software, such as application servers and identity management (Oracle, 2007).

The company doesn't disclose its revenue share by application type. But since PeopleSoft and Siebel had revenues of \$2.27 billion and \$ 1.34 billion, respectively, before they were acquired (Waxer, 2006), it can be assumed that Oracle generates more than 30% of its total revenues of \$14.38 billion (2006) from ERP and CRM products.

### **3.2.3.4 SAP**

As one of the few leading European software companies, SAP, founded in 1972, is the market leader in the ERP market with a market share of more than 40% (AMR Research, 2006). SAP's revenues of \$9.4 billion (2006) are almost completely generated in this segment of the market, although the company has a relatively high share of service revenues (SAP, 2006).

SAP's product range covers most typical ERP applications, as well as some more specialized fields such as product lifecycle management and supplier relationship management. The company traditionally sells mostly to larger customers, boasting an impressive cus-

tomers list. Nevertheless, SAP is trying to enter the SMB market with its SAP Business One range of products (SAP, 2007).

SAP's main strength is probably its long experience in building fully integrated ERP solutions that are based on an enterprise-wide, fully unified data structure. However, over the last few years SAP has developed additional tools and technology platforms that enable SAP's applications to become more modular and flexible (Agassi, 2006).

#### **3.2.3.5 Intuit**

Intuit, founded in 1983, sells financial management and tax software for small businesses, consumers and accountants and had 2006 revenues of \$2.34 billion. Its main product lines are the TurboTax tax management software, QuickBooks accounting software and the Quicken line of personal finance software (Intuit, 2007).

#### **3.2.3.6 Adobe Systems**

Adobe systems, founded in 1982, is focused on solutions for creative content production. Its flagship products (Photoshop, Premiere, Flash, Dreamweaver) are predominantly used by creative professionals who edit photos, videos and websites. Other products have a broader target market: Adobe's Acrobat product line is the dominant solution for electronic document production and distribution (Graham 2006). Adobe had revenues of \$2.45 billion in 2006 (Adobe, 2006).

#### **3.2.3.7 Autodesk**

Autodesk sells a range of products for engineering and design. Its AutoCAD product for computer-aided design is the leader in this specialized market. Autodesk's revenues in 2006 were \$1.52 billion (Autodesk, 2006).

#### **3.2.3.8 The Sage Group**

The Sage Group, headquartered in the United Kingdom, is a supplier of ERP solution and services with 2006 revenues of £935.6m (= \$1.8 billion). Sage has acquired a number of smaller ERP companies all over the world, totaling 16 companies between 2003 and 2006 (Sage, 2007). As a consequence, Sage offers a number of different product lines that mainly target the SMB market.

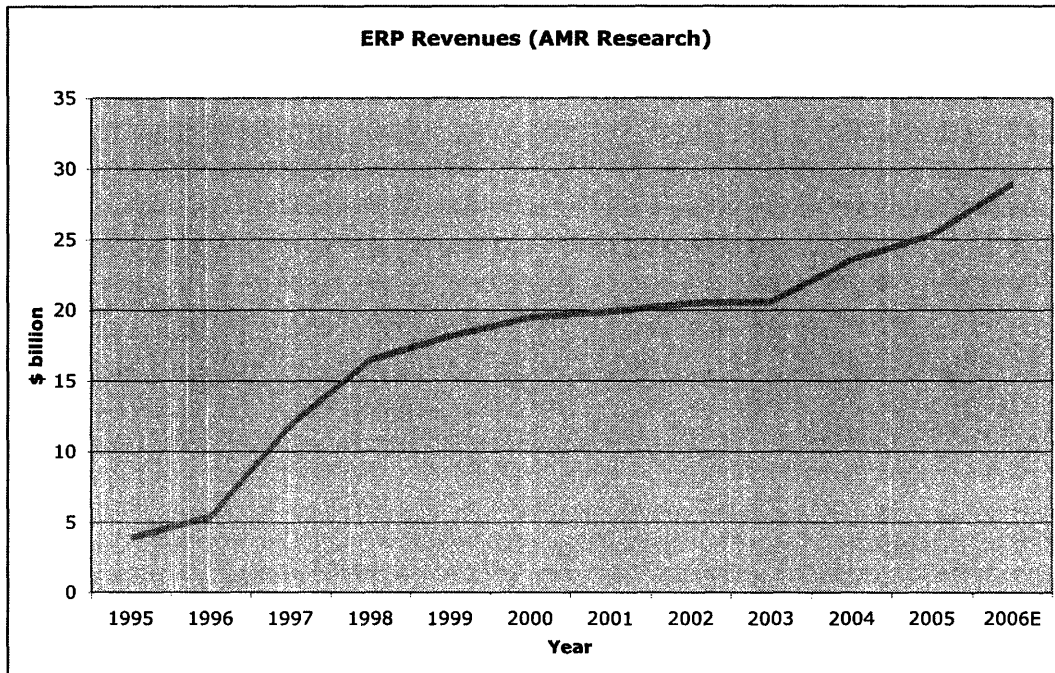
#### **3.2.3.9 Infor / SSA Global**

SSA Global, the fifth-largest ERP vendor (AMR Research, 2006), was recently acquired by privately held Infor, which is owned by a group of private equity firms. According to the company, the combined entity had 2006 revenues of \$2.1 billion. The company offers a wide range of business solutions, predominantly for customers in manufacturing, distribution and services (Infor, 2007).

### **3.2.4 Market Development**

The market for packaged enterprise applications experienced rapid growth in the 1990s until 2000. After that, the slowdown in the world economy caused a stagnation of the market between 2001 and 2003. After 2004, the market regained some of its growth dynamics, although on a much lower level than in the 90s.

The following graph shows the development of the ERP (enterprise resource planning) market segment that accounts for approximately 40% of the enterprise applications market (AMR Research, 2006).



**Figure 1: ERP revenues 1995-2006, US \$ billion (AMR Research, 2006)**

The most defining trend of the last few years in the enterprise software market was a wave of consolidation (Bokhari, 2006). The systematic acquisition of smaller software vendors by larger consolidators is by no means a new phenomenon in the software industry. The maturing mainframe software segment experienced similar concentration in the 1980s and early 1990s, led by firms like Computer Associates or Dun & Bradstreet Software (Campbell-Kelly, 2004).

Of the nine companies listed above, three (Oracle, Sage, Infor) clearly focus their strategy on the acquisition and integration of smaller enterprise software vendors. The other six companies also made major acquisitions that in some cases (e.g. Microsoft's acquisition of Great Plains in 2000 and Navision in 2002; Hoover, 2007) actually laid the groundwork for their entry into new market segments.

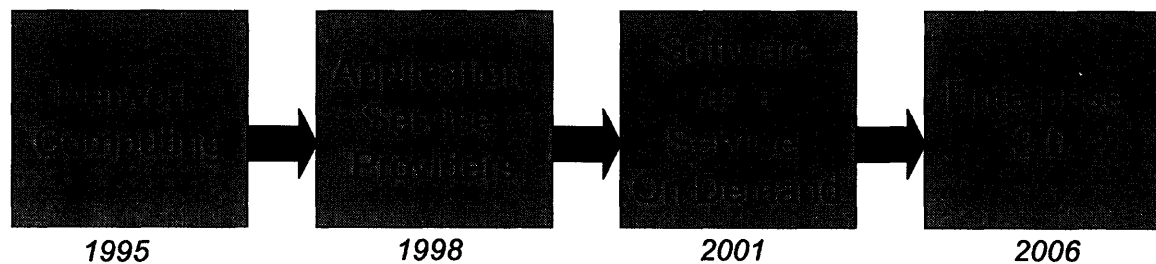
## 4 Web-based Enterprise Software

### 4.1 Historical Background

In the era of the mainframe computer, enterprise software was often sold as a part of an overall solution, consisting of hardware, software, and services (Cusumano, 2004). This era was dominated by IBM who was one of very few companies able to supply all necessary elements.

The Personal Computing brought the era of client-server computing and of independent vendors of enterprise applications. Companies such as SAP and Oracle built their growth based on this new paradigm that separated the various necessary elements of a solution. Customers would buy hardware, infrastructure software and application software from different vendors, integrating the individual pieces themselves or with the help of a system integrator. The PC, running a full client software program, was the key user interface to enterprise applications, advancing the dominance of Microsoft and Intel.

Over the last 15 years, there have been several attempts to replace this model with a new paradigm. The following figure illustrates the sequence of concepts that focus on the common goal of reducing the importance of the PC and establishing a new model with different economic qualities.



**Figure 2: Historical development of alternative paradigms for software deployment**

In the mid to late 1990s, several companies tried to sell so-called “Network Computers” (NC), simple terminal-like client computers that were used to access centralized servers that ran the actual applications. NC vendors, including IBM, Sun Microsystems and Oracle, emphasized the reduced costs, improved security and greater flexibility of network computing. However, since NCs had many limitations when compared to PCs, they never caught on in the market. The rapidly declining costs of standard PCs limited the economic appeal of cheap NCs further. (Briody, 1999)

The late 1990s saw the rise and partial fall of Application Service Providers (ASPs). ASPs provide traditional on-premise software as a hosted service to customers over a network, often the public Internet. Customers don’t have to operate their own server in order to use an enterprise application, but rather outsource this task to the ASP. ASPs typically own the necessary software licenses and bill customers on a “per use” basis or based on a recurring subscription fee. However, ASPs are usually not the actual software vendors, but concentrate on the operation of the necessary infrastructure. The applica-

tions used in ASP setups are typically proprietary software packages, often using a proprietary client program.

The ASP wave can generally be seen as a precursor to the currently popular “Software as a service” movement (see below, section 4.3.3), but the first generation of ASPs stayed far below expectations. The ASP market was estimated to have generated revenues of around \$2 billion in 2004, far below the predictions a few years earlier (McDougall, 2005).

Around 2001, the notions of “Software as a Service” and “On Demand Computing” started to become popular. The on-demand concept was strongly driven by a major marketing campaign started by IBM in 2002 (Junnarkar, 2003). While the on-demand notion focused more on computing infrastructure, “Software as a Service” (SaaS) applied the same basic ideas to applications. Analyst firm Gartner Inc. defines SaaS as “software that is owned, delivered and managed remotely by one or more providers. The provider delivers an application based on a single set of common code and data definitions, which are consumed in a one-to-many model by all contracted customers, at any time, on a pay-for-use basis, or as a subscription based on usage metrics.” (Lheureux, 2006).

In 2006, Andrew McAfee introduced the term “Enterprise 2.0” in an article in the MIT Sloan Management Review. He used this notion to describe a “new wave of business communication tools including blogs, wikis and group messaging software” (McAfee, 2006). The driving force behind these new wave were concepts found in the so-called Web 2.0 movement (O’Reilly, 2005). In essence, Enterprise 2.0 extends the older SaaS concept with a business-oriented use of new types of collaborative software.



The following table summarizes the main characteristics of these different generations of network-centric applications:

	<b>Network Computing</b>	<b>Application Service Providers</b>	<b>Software as a Service</b>	<b>Enterprise 2.0</b>
<b>Client environment</b>	“Thin client”: network terminal based on specialized hardware	PC, usually with locally installed client software (“Fat client”)	Web browser, running on PC or other device	Web browser, running on PC or other device
<b>Main driver for adoption</b>	Cost reductions	Cost reductions	Cost reductions, flexibility, time to value	Increased need for collaboration
<b>Application Hosting (typical case)</b>	Internal at user company	Externally at ASP, single tenant model (dedicated servers / software instances)	Externally, typically by software vendor. Multi-tenant model (same code for all users)	Not defined
<b>Type of software</b>	Simple, task focused.	Complex business applications, based on existing on-premise software	Predominantly commoditized business applications, based on software specifically developed for SaaS	Collaborative applications
<b>Key business model for vendors</b>	Hardware sales (client machines and servers)	Pay-per-use or recurring subscription fees.  New type of intermediary infrastructure service (between software vendor and user)	Typically recurring subscription fees.  Vertically integrated model (software vendor captures most of the value).	Subscription fees or licenses; products often based on open source software, hence free.

**Table 1: Main characteristics of network-centric application generations**

## 4.2 Web 2.0 and Enterprise Software

In 2004, the notion of “Web 2.0” was introduced at a conference organized by publishing company O’Reilly Media. Lacking a clear definition, Web 2.0 quickly became an umbrella term for a whole generation of new web-based applications and services.

Several successful websites have been described as being part of the Web 2.0 wave, including video sharing site YouTube, online encyclopedia Wikipedia, and photo sharing site Flickr. Furthermore, new application types with a collaborative focus, such as blogs and wikis, are generally subsumed under the Web 2.0 moniker (O’Reilly, 2005).

Only in 2006, O’Reilly Media CEO Tim O’Reilly, often credited as the inventor of the term, tried to come up with a compact definition:

“Web 2.0 is the business revolution in the computer industry caused by the move to the internet as platform, and an attempt to understand the rules for success on that new platform. Chief among those rules is this: Build applications that harness network effects to get better the more people use them.” (O’Reilly, 2006)

To some extent, many web-based enterprise software products fit this definition. Quite obviously, the Internet is the platform and basic infrastructure for these products.

It is more difficult to find evidence for network effects. A network effect is a property of a system that lets it become more valuable to everybody the more people use it (Shapiro and Varian, 1998). Many web-based software products offer mechanisms for the sharing of information with other users. For example, users of Google Docs & Spreadsheets, one of Google’s online applications, can invite other Internet users to collaborate on documents, thus increasing the application’s number of registered users. Many web-based applications also offer open programming interfaces that can be used to link independent programs, thereby generating useful new combinations of functionality. Furthermore, some of the technical properties generally associated with Web 2.0, such as AJAX (asynchronous JavaScript and XML; Graham, 2006) are often used in web-based enterprise software.

However, many of the most successful applications do not exhibit strong network effects. For instance, Salesforce.com as one of the leading vendors of web-based software originally did not use any features that would establish connections between its corporate customers, and therefore its early success wasn’t based on network effects. Only in its latest developments, such as the application marketplace AppExchange, Salesforce.com tries to establish network effects through an open platform (Salesforce.com, 2007b).

In summary some of the defining Web 2.0 characteristics do not really apply to business applications of the type examined here. Therefore, Web 2.0 is not a particularly useful or relevant concept for the understanding of web-based enterprise software.

### 4.3 Characteristics of Web-based Enterprise Software

Section 3.1 already discussed the basic definition of “web-based enterprise software” that this thesis will use.

The following characteristics are typical for web-based enterprise software:

- Web-based user interface
- Hosted application deployment
- “Software as a service” business model
- Service-oriented Architecture for integration

These aspects will be explained in more detail in the following sections.

#### 4.3.1 Web-based User Interface

Web-based enterprise software uses standard web browsers (such as Microsoft Internet Explorer, Firefox, Safari, Opera) as the front-end infrastructure, as opposed to proprietary, locally installed client software.

Typically, a web-based application can be used with any of these standard browsers, as long as the user has a relatively recent version of the browser program that supports current technical standards.

This has the following consequences for users and vendors of web-based applications:

<i>Web-Based User Interface Consequences for Users</i>	
<b>Advantages</b>	<b>Disadvantages</b>
<ul style="list-style-type: none"> <li>• <b>Cost reduction for installation and maintenance:</b> There is no cost for the installation and maintenance of proprietary client software on users’ computers. Web browsers are a standard element of a modern PC’s basic pre-installed software, so almost all currently used PCs are likely to have an adequate web browser already.</li> <li>• <b>Training cost reduction:</b> The use of a standardized, already familiar interface reduces user training times dramatically. Most PC users nowadays can be assumed to be familiar with web browsers and the user interface conventions that websites commonly use. Web-based enter-</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Reduced “richness” of user interface:</b> Even modern AJAX-based browser interfaces can’t match the rich interactivity and immediacy of a locally installed PC application. This has some potentially disadvantageous consequences for productivity.</li> <li>• <b>Not suitable for some applications:</b> The still limited bandwidth between clients and servers limits the usefulness for some very data-rich application types, such as multimedia production applications (video editing, image editing, etc.) or computer aided design (CAD). Although there are some pioneering web-based applications in this</li> </ul>

<p>prise software builds on this stock of familiarity.</p> <ul style="list-style-type: none"> <li>• <b>Reduced OS dependency:</b> To some extent, the client PC's brand of operating system (OS) becomes irrelevant. Most web-based software works in exactly the same way on a Windows PC, a Macintosh or a Linux machine. Platform-independent browser software such as Firefox further reduces the dependency on a specific OS.</li> </ul>	<p>space, they are by far not strong enough for professional use. Therefore, "going all web-based" is not an option for many users.</p> <ul style="list-style-type: none"> <li>• <b>No complete OS independence:</b> There are some applications where the client OS still is relevant because there is a piece of locally installed software that interacts with the web-based application.</li> </ul>
---	---

<i>Web-Based User Interface Consequences for Vendors</i>	
<b>Advantages</b>	<b>Disadvantages</b>
<ul style="list-style-type: none"> <li>• <b>Reduction of development costs:</b> Vendors do not have to develop a separate client application, thereby reducing total development costs massively.</li> <li>• <b>Reduced OS dependency:</b> This point also applies to vendors. They don't have to decide for which client platform to write their application, and there are no additional costs for porting.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Different user interface paradigm:</b> Although browsers have become more capable over the last few years, there are still significant differences in the way user interfaces are structured compared to PC-based clients. This requires a complete re-thinking of user interfaces for existing applications and retraining of developers.</li> <li>• <b>Limited compatibility:</b> Although modern web browsers stick to a set of well-defined technical standards (such as XHTML, CSS2, JavaScript), there are still many subtle differences between browser families and versions that reduce compatibility. Developers often spend a lot of time adapting their web UIs for different browsers.</li> </ul>

### 4.3.2 Hosted Application Deployment

Traditional enterprise software is installed locally on the customer's premises, typically in a client/server-setup. A server machine that is positioned inside the customer's net-

work runs the server component, and client software on users' PCs provides the user interface.

In contrast, web-based enterprise software uses a different approach: The server component doesn't run on a server machine owned and operated by the customer, but on infrastructure owned, operated and managed by the software vendor (or, less frequently, a third party). Customers access the application remotely over the Internet.

In most cases, vendors use a so-called "multi-tenant" platform, as opposed to the "single tenant" hosting used by application service providers (ASPs). In a multi-tenant setting, several customers share a common application server (Maoz, 2006). The application itself is built in such a way that each customer can customize his instance of the software, and of course security measures prevent one customer from seeing another's data. The individual customer perceives the application to be running on an isolated server.

<i>Hosted Application Deployment Consequences for Users</i>	
<b>Advantages</b>	<b>Disadvantages</b>
<ul style="list-style-type: none"> <li>• <b>Flexibility and speed of deployment:</b> Since no local installation is necessary for a hosted application, users can start to use an application very quickly. For some simpler offerings, it is possible to start using the software literally immediately after a short registration process.</li> <li>• <b>No infrastructure investments:</b> The customer doesn't have to make an upfront investment in server hardware, installation services and additional software (such as database systems), thus reducing the capital expenditure for a new application. The only necessary infrastructure on the customer side are adequate client PCs with web browsers and a broadband Internet connection.</li> <li>• <b>Reduced maintenance costs:</b> Since the server infrastructure is operated by the vendor, the customer saves on maintenance expenditures and IT personnel.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Availability concerns:</b> The availability of the application depends on the uninterrupted operation of the broadband Internet connection. Since a typical Internet connection between a client and a server runs through the networks of several operators, there are multiple points of failure, most of which are outside of the control of both the customer and the software vendor. The customer entrusts the software vendor with the operation of the server infrastructure and therefore depends on the vendor's ability to provide uninterrupted service. This problem is often mitigated by service level agreements (SLAs) that guarantee a defined percentage of availability.</li> <li>• <b>Confidentiality and security concerns:</b> Since a hosted application stores its data on the vendor's servers, the customer in fact entrusts the vendor with safeguarding proprietary and potentially very sensitive information. This confidentiality issue is a major hurdle to the adop-</li> </ul>

	tion of web-based enterprise software (Vincent et al., 2005).
--	---

To address users' concerns, many vendors of web-based software offer alternative deployment models that enable customers to run an application "on premise". There are three basic variations:

- Some software companies offer classic installable versions of their products that can run on a customer's own servers.
- Hardware appliances are increasingly gaining popularity (Hein, 2007). In this model, the vendor supplies a completely pre-configured server machine with all necessary software components pre-installed. This "appliance" server is integrated locally into a customer's network and often managed by the customer's IT staff (Prentice, 2007).
- A very recent development are so-called "virtual appliances" that make use of virtualization technology. A virtual appliance is a fully pre-configured server environment that doesn't run on dedicated hardware, but on a virtual machine on a larger server. This has the advantage of reduced costs compared to the hardware appliance model and increased control for the customer's IT staff (Babcock, 2006).

<i>Hosted Application Deployment Consequences for Vendors</i>	
<b>Advantages</b>	<b>Disadvantages</b>
<ul style="list-style-type: none"> <li>• <b>Cost reduction through simplicity of deployment:</b> One of the major cost factors in traditional software deployment is the management of upgrades and support for different versions that customers use. This problem doesn't exist in the hosted model, since the vendor can immediately upgrade the software on the hosting infrastructure.</li> <li>• <b>Server platform independence:</b> Vendors do not have to develop for different server platforms (such as Windows, Solaris, Linux), but can select the platform that they feel most comfortable with.</li> <li>• <b>Economies of scale:</b> Vendors of web-based software typically run</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Added complexity:</b> Operating a hosting infrastructure is a complex task that not all software vendors are prepared for. The characteristics of a 24/7 hosting operations are very different from those of a software development organization, and therefore management resources could be strained.</li> <li>• <b>Added risk of liability:</b> Vendors have to guarantee a certain level of availability for applications and also confidentiality for users' data. Both pose a risk that vendors of classical software do not have to the same extent.</li> <li>• <b>Requirement to offer alternative deployment methods:</b> As ex-</li> </ul>

<p>infrastructure for a large number of customers. This results in economies of scale that reduce overall costs for customers. This particularly applies in the multi-tenant model, where all customers use the same code and configuration of the software.</p> <ul style="list-style-type: none"> <li>• <b>Smaller need for professional services:</b> Since most hosted solutions are deployed as a standardized offering with limited customization, the need for professional services is smaller than for traditional software. Companies such as Salesforce.com or RightNow Technologies generate only a 10-15% share of revenues from professional services (Salesforce, 2006; RightNow, 2006), compared to traditional vendors like SAP that have a 30% or bigger share of professional services revenue (SAP, 2006).</li> </ul>	<p>plained above, not all customers feel comfortable with hosted applications. Vendors therefore decide in many cases to offer alternative deployment models, thereby reducing the advantages of the hosted model.</p>
---	--

### 4.3.3 “Software as a Service” Business Model

Traditional enterprise software is typically sold based on a perpetual license with additional annual maintenance fees. The customer pays a certain license fee upfront when first installing the software and then has to pay a maintenance fee every year (usually between 15-20% of the upfront license amount) to ensure continued support and software upgrades from the vendor. In some cases, vendors charge additional license amounts for major upgrades of their software products.

Vendors of web-based enterprise software typically charge based on a different model: customers pay no upfront fees, but rather a subscription fee (recurring monthly or annually) per user that covers the complete use of the software for the relevant time span, including the operation of server infrastructure, a defined level of support and software upgrades. For instance, Salesforce.com’s fees start at \$65 per user per month for its CRM product. This business model is often called “Software as a Service” (SaaS).

*“Software as a Service” Business Model  
Consequences for Users*

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>• <b>Reduced investment costs:</b> There is no upfront license fee. Therefore, the overall initial investment decreases dramatically compared to traditional software.</li> <li>• <b>Predictability of fees:</b> The recurring costs are easy to calculate, since the subscription fees include all necessary elements for the operation of the application.</li> <li>• <b>Reduced “sunk costs”:</b> When the customer decides not to use the application anymore, he typically can cancel the contract with the software vendor on relatively short notice. In contrast, in the traditional model there is often no way to recoup upfront license expenses for an application that isn’t used anymore.</li> <li>• <b>Flexible rollout, no unused “seats”:</b> Companies can typically rent SaaS applications for every user individually. This enables a flexible rollout in a company, starting with only a few users and growing as appropriate. A typical problem in classical enterprise software is the phenomenon of unused “seats”, because vendors prefer to sell large blocks of “seat” licenses to customers that are unsure about how many people will use the software (Moore, 2005). This problem doesn’t apply to SaaS.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Unknown cost for total application life:</b> In the case of traditional software licenses, the bulk of payments is due at the beginning when the user buys the license. This doesn’t apply to SaaS, where the customer has to pay every month. If a customer uses an application for a very long time without upgrades, the total cost of a SaaS solution could exceed that of a licensed software product. Furthermore, changes in the SaaS vendor’s pricing after the contract duration aren’t predictable.</li> <li>• <b>Potential “Lock-in”:</b> Vendors use very different contract lengths for their SaaS offerings. While some contracts can be canceled every month, others have an annual or even longer duration.</li> <li>• <b>Increased leverage for vendor:</b> Since SaaS vendors own the software product and provide continuous service, their leverage over customers is greater than that of traditional software companies. For instance, a vendor can turn off service for a non-paying customer immediately to enforce payment (Bona, 2004).</li> </ul>



<i>“Software as a Service” Business Model Consequences for Vendors</i>	
<b>Advantages</b>	<b>Disadvantages</b>
<ul style="list-style-type: none"> <li>• <b>Predictable stream of recurring revenues:</b> SaaS vendors receive payments from customers every month (typically), which enables them to plan revenue development more reliably than with traditional license-based models.</li> <li>• <b>Reduced discount pressure:</b> Sophisticated buyers of enterprise software know that the best time to strike a license deal is at the end of the quarter, when vendors have to make their numbers. Many enterprise software licenses are therefore sold with large discounts. This mechanism doesn't exist in the same way in the SaaS world, since the vendor recognizes revenue over a long time and therefore doesn't have an incentive to sell licenses cheaply just to make quarterly numbers.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Delayed revenue recognition:</b> Since subscription revenues are recognized over time and not like license sales at the time of the first delivery of the product, revenue growth for SaaS companies is slower.</li> <li>• <b>Reduced gross margins:</b> Since the operation of a hosting infrastructure is a key element for a SaaS offering, the corresponding costs reduce the overall gross margin that a vendor can earn.</li> </ul>

#### **4.3.4 Service-oriented Architecture for Integration**

A major consideration for any kind of enterprise application is the need for integration with other applications. For instance, an accounting application needs to exchange data with the payroll system, the sales force management system and inventory management. One of the key advantages of integrated ERP systems (such as the ones sold by SAP and Oracle) is the high degree of integration between functional modules, typically enabled by a common database.

For web-based applications, this level of integration is quite difficult to achieve. For instance, a customer might use a hosted CRM application from one vendor, an internal financial application from another and a hosted project management product from yet another. It would be desirable to have all these applications “talk” to each other, but since they reside on different servers in different networks, running on different operating systems, this is by no means easy to achieve. Even in a traditional setup where all applica-

tions run inside a firm's own network, integration is difficult, but it gets even more complicated in a highly distributed SaaS scenario.

A potential solution for this problem is a so-called service-oriented architecture (SOA). SOA uses XML-based web services to interconnect applications that run on entirely different hardware platforms and operating systems. The applications communicate using standardized XML formats that are transmitted over a TCP/IP network. This approach is often used in heterogeneous system environments inside a company, but it can also be applied to externally hosted applications.

<i>Service-oriented architecture for integration Consequences for Users</i>	
<b>Advantages</b>	<b>Disadvantages</b>
<ul style="list-style-type: none"> <li>• <b>Flexibility:</b> In principle, the “loose coupling” of SOA-based application interfaces allows companies to integrate internal and external applications very flexibly. There are, however, many details in specific integration cases that can reduce this flexibility.</li> <li>• <b>Leverage legacy applications:</b> Many companies use a SOA layer that sits on top of an existing legacy application and exposes functionality to other applications via web services. This enables companies to use existing functionality in combination with new web-based applications.</li> <li>• <b>Increased platform independence:</b> SOA works across different OS platforms and even network types, so customers can integrate very heterogeneous systems.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>New skill set:</b> Internal developers and IT operation staff have to learn new skills to work with SOA environments. This can be costly and time-consuming.</li> <li>• <b>Stability and performance concerns:</b> Since a SOA-based integration depends on many more components than a local integration on the same machine, overall stability and performance are presumably not the same.</li> <li>• <b>Limited maturity of infrastructure and services:</b> Because SOA is a comparatively new philosophy, most infrastructure software products, development tools, methodologies and the professional services that can be obtained from outside consulting firms do not yet have the same degree of maturity as more established integration methods.</li> </ul>

<i>Service-oriented architecture for integration Consequences for Vendors</i>	
<b>Advantages</b>	<b>Disadvantages</b>
<ul style="list-style-type: none"> <li>• <b>Flexibility:</b> SOA enables vendors to integrate various systems very quickly.</li> <li>• <b>Cost savings through reuse:</b> SOA enables software companies to reuse existing functionality, independently of where and on what kind of platform this functionality exists. For instance, many companies now integrate mapping features into their products that use features offered by Google Maps or other mapping providers.</li> </ul>	<ul style="list-style-type: none"> <li>• <b>Increased complexity:</b> SOA introduces multiple points of failure into the communication between two software components, leading to increased complexity for development, debugging and operations.</li> <li>• <b>Performance issues:</b> Since the XML-based protocols used by SOA introduce considerable overhead, performance of SOA-based applications can be an issue.</li> <li>• <b>Evolving standards:</b> The basic web service standards for SOA are still evolving, leading to some degree of insecurity in the development of web-based applications (Vincent et al., 2005).</li> </ul>

#### 4.4 Summary

Web-based enterprise software has a number of specific characteristics, both in economic and technical dimensions. For customers, increased flexibility, reduced investment costs and reduced dependence from technical platforms are the most positive traits. On the negative side, data confidentiality, reliability and integration are the most significant concerns.

For vendors, web-based applications provide very different economics and require a new set of technical skills compared to traditional software. There are also a number of new risks and unsolved problems that vendors face.

## **5 Market Structure**

The following chapter tries to characterize the current state of the market for web-based enterprise software, based on a sample of vendors.

Section 5.1 will briefly explain the sample structure and sources. In section 5.2, a framework for the classification and analysis of web-based enterprise applications is established, based on the characteristics found in the sample. In section 5.3, the sample of vendors is analyzed in detail using the framework. Section 5.4 examines a number of typical vendors and their strategic approaches with a series of short case studies. Section 5.5 analyzes the currently available data about customer adoption of the new model for web-based software. Finally, section 5.6 will examine if there is evidence for a disruptive character of web-based applications, and section will 5.7 look for emerging dominant designs in this market.

### **5.1 Sample Description**

The sample of companies that sell web-based enterprise software was collected between January 07 and March 07 from the following sources:

- Application Software Industry company list (Yahoo Finance, 2007).
- A variety of industry publications (Red Herring magazine, Information Week) and online sources (TechCrunch, 2007; VentureBeat, 2007; Alarm:clock, 2007).
- Participant and speaker lists from conferences that deal with SaaS, on demand and Web 2.0 concepts:
  - Software 2006 conference, Santa Clara, April 4-5, 2006
  - Office 2.0 conference, San Francisco, October 11-12, 2006
  - SIIA OnDemand: San Jose, November 8-9, 2006
  - Under the Radar – Why Office 2.0 matters, Mountain View, March 23, 2007
  - SaaScon, Santa Clara, April 17-18, 2007
  - Enterprise 2.0 conference, Boston, June 18-21, 2007
- Google search for relevant keywords.
- Author's own experience and discussions with people who are active in the software industry.

The following criteria were used to include a given software company in the sample:

- The company must offer a business application that is typically used internally in enterprises. Software products that focus purely on the maintenance of web pages, online marketing or similar externally focused purposes were excluded from the sample.

- The company must offer a software product that is fully usable with a web-browser. No locally installed client software must be required (except for standard freely available run-time environments like Adobe Flash and Java).
- The company must offer a hosted version of the application. However, if a company offers alternative deployment methods in addition to the hosted version, it was included in the sample.
- The company must provide a product that is offered publicly. Products that are still in closed Alpha or Beta tests were not included in the sample.

By applying these criteria, the original sample size of 149 companies was reduced to 108 companies meeting the criteria.

The information used in the sample database is based on companies' own public websites and additional information from the sources mentioned above.

## **5.2 Framework for the Classification of Web-based Enterprise Applications**

The previous chapter explored the general characteristics of web-based enterprise applications. In order to become more precise and specific, this section will introduce a framework for the classification of web-based applications. The goal is to show in which fields and using which technical and financial models web-based applications are currently being offered.

Since many vendors of web-based applications use combinations with other forms of deployment and revenue models, the adjacent models will also be listed here.

The criteria used here are based on the sample of companies and applications discussed in more detail in section 5.3.

### **5.2.1 Application type**

A first important classification factor is the application type, i.e. the purpose of the application. Section 3.2.2 already listed the most common application types for traditional enterprise software, and it is not surprising that there is a large overlap to web-based enterprise applications. However, there are types that are new and specific to web-based software, and there are also types that do not yet appear in the form of web-based products.

The following tables also list some typical examples of companies offering products in the specific area.

### 5.2.1.1 Horizontal Enterprise Processes

<b>Horizontal Enterprise Processes</b>		
<i>Application Type</i>	<i>Description</i>	<i>Examples</i>
CRM (Customer Relationship Management)	Management and analysis of information about customers, planning of marketing and sales campaigns, collaboration with customers and third parties.	Salesforce.com, Oracle/Siebel OnDemand
Sales Force Management	Support of sales processes including sales lead management, sales forecasting, compensation management.	Salesforce.com, Xactly
ERP (Enterprise Resource Planning)	Integrated planning and management of corporate resources, including finance, inventories, manufacturing capacity and so on.	NetSuite, Workday
Business Process Management (BPM)	Operational management and monitoring of business processes	Adaptive Planning
Business Intelligence	Analysis of corporate data	Crystalreports.com (Business Objects), LucidEra
Finance	Finance-specific applications, e.g. accounting, billing, payment	Netsuite, Intacct, Intuit QuickBooks Online Edition
HR	Human resources management applications, compensation management, talent management	Taleo, SuccessFactors
Supply chain management	Collaboration with suppliers, management of supply flows	Panthius

**Table 2: Application types for horizontal enterprise processes**

### 5.2.1.2 Personal Productivity

<b>Personal Productivity</b>		
<i>Application Type</i>	<i>Description</i>	<i>Examples</i>
Calendar	Calendar and time management	Google Calendar, 30 Boxes
e-Learning	Computer-based learning; creation and management of teaching materials	Plateau, Simulat

e-Mail	Electronic mail	Google Gmail, Microsoft Windows Live Mail
Presentation	Creation, management and sharing of presentation material (similar to PowerPoint)	Spresent, Think Free
Spreadsheets	Creation and editing of spreadsheets	Google Spreadsheets, ThinkFree
Word processing	Creation and editing of text documents	Zoho Writer, Google Docs

**Table 3: Application types for personal productivity**

### 5.2.1.3 Communication and Collaboration

<b>Communication and Collaboration</b>		
<i>Application Type</i>	<i>Description</i>	<i>Examples</i>
Blog	Software for the creation and management of online weblogs (blogs)	BlogTronix Near-Time
Conferencing	Real-time communication with remote partners, using text, voice and/or video	Webex, Thinkature
Content Management	Structured creation, sharing and management of documents and other digital content	Koral, Cogenz
Portals	Portal software that consolidates information individually for each user	Google, Oracle
Project management	Management of tasks, milestones and resources for distributed project teams	37signals, Microsoft Office Live, Smartsheet.com
Social Networking	Online networking for people inside and outside of a company	CollectiveX, LeverageSoftware
Social Tagging	Information management using collaborative tagging methods	ConnectBeam, Diigo
Team collaboration	Information exchange and coordination of distributed teams	37signals, Central Desktop, Microsoft Office Live
Wiki	Collaborative information creation and management	SocialText, BrainKeeper

Workflow	Structured sequential treatment of documents or other digital objects	Approver.com, EchoSign
----------	---	------------------------

**Table 4: Application types for communication and collaboration**

#### 5.2.1.4 Vertical and Specialized Applications

<b>Vertical and Specialized Applications</b>		
<i>Application Type</i>	<i>Description</i>	<i>Examples</i>
Field Service Management	Management of and communication with field support staff	Astea International
Healthcare	Healthcare-specific applications, such as patient record management	Purkinje
Imaging	Management and sharing of visual material	Adobe, Idee
Professional Services Automation	Management of professional services organizations; time sheet management, billing	AutoTask, OpenAir
Site management	Real estate management for mobile telephony operators	Siterra

**Table 5: Application types for vertical and specialized applications**

#### 5.2.1.5 Infrastructure and Application Development

<i>Application Type</i>	<i>Description</i>	<i>Examples</i>
Application Development	Tools and frameworks for the development of web-based applications	Etelos, AppExchange Teqlo
Database	Relational database for online use	DabbleDB, Intuit QuickBase
Application Infrastructure	Run-time environments and middleware for hosted applications	Adobe Web Services, OpSource
Mobile Access	Data and application access via PDAs, Smartphones and other mobile devices	IBM, Microsoft
RSS	Distribution of digital content in RSS (Really Simple Syndication) format	NewsGator
Storage	Online storage for documents and other digital goods	DropSend, OmniDrive

**Table 6: Application types for infrastructure and application development**



## 5.2.2 Revenue model

Vendors of web-based applications currently use the following revenue models:

<i>Model</i>	<i>Subtypes</i>	<i>Description</i>
Subscription	<ul style="list-style-type: none"> <li>• Monthly</li> <li>• Annually</li> </ul>	<p>Recurring payment of a subscription fee. Sometimes in combinations: monthly payment, but annual contract.</p> <p>In many cases, subscription fee depends on features requested by user (different versions of the software)</p>
Free	<ul style="list-style-type: none"> <li>• Limited free trial</li> <li>• Free basic version</li> <li>• Free beta version</li> <li>• Free but not free (bundled)</li> </ul>	<p>Customer isn't charged at all; often used for basic versions of an application or in beta phase.</p>
Advertising	<ul style="list-style-type: none"> <li>• Display ads</li> <li>• Keyword-based ads</li> </ul>	<p>Application is free, but users have to accept advertising that is included in the user interface.</p>
Upfront license	<ul style="list-style-type: none"> <li>• One-time only</li> <li>• With annual maintenance fees</li> </ul>	<p>Classic revenue model for enterprise software: User pays a license fee at the beginning, in many cases additionally an annual maintenance fee.</p>
Pay-per-use	<ul style="list-style-type: none"> <li>• Per transaction</li> <li>• Percentage of transaction volume</li> <li>• Pay per usage time</li> </ul>	<p>User pays only for actual usage. Calculation can be based on a variety of metrics.</p>
Professional services	<ul style="list-style-type: none"> <li>• Consulting</li> <li>• Support</li> <li>• Installation</li> </ul>	<p>Vendor charges for professional services in addition to other revenues or as the only source of revenue.</p>

**Table 7: Revenue models**

Many of these models are subject to discounts for larger users. However, almost no vendor openly publishes discount schemes.

### 5.2.3 Server Deployment Model

The server side of a web-based application is most typically hosted remotely. However, there are some other models that are offered in combination or as alternatives. These models were already explained in section 4.3.2 and are here only listed for completeness.

Server Deployment Models	Subtypes
Remotely hosted service	<ul style="list-style-type: none"> <li>Operated by software vendor</li> <li>Operated by third party</li> </ul>
Locally installed software	<ul style="list-style-type: none"> <li>Different operating systems and middleware environments</li> </ul>
Hardware appliance	
Virtual appliance	

**Table 8: Server deployment models**

### 5.2.4 Client Deployment Model

Although web-based applications use a standard web browser as the user interface, there are some subtle variations that can restrict the universal use of a specific application. In some cases, users might have restrictions for some of these models due to technical limitations or security concerns.

Client Deployment Models	Subtypes
All standard web browsers compliant with W3C standards	<ul style="list-style-type: none"> <li>Static HTML</li> <li>Dynamic HTML / AJAX</li> </ul>
Specific type or version of web browser (e.g. only Microsoft Internet Explorer 6.0 and higher)	<ul style="list-style-type: none"> <li>various</li> </ul>
Freely available run-time environment in combination with web browser	<ul style="list-style-type: none"> <li>Adobe Flash</li> <li>Java</li> <li>Microsoft .NET</li> </ul>
Required web-browser plug-ins or toolbars	<ul style="list-style-type: none"> <li>Platform-independent (e.g. Firefox plug-ins)</li> <li>Platform-dependent (e.g. Microsoft ActiveX plug-ins)</li> </ul>
Optional locally installed client software	<ul style="list-style-type: none"> <li>Local client that enables enhanced functionality</li> <li>Local client that enables offline use when no Internet connection is available</li> </ul>

**Table 9: Client deployment models**

### 5.2.5 Integration Model

Enterprise applications are rarely stand-alone, but have to interchange information with other systems. The type of integration model that a vendor offers can in many cases determine in which type of environment a specific application can be used.

Integration Model	Subtypes
Fully programmable platform	<ul style="list-style-type: none"> <li>• With hosting on vendor's infrastructure</li> <li>• Without hosting</li> </ul>
Open application programming interfaces (APIs)	<ul style="list-style-type: none"> <li>• Based on open standard</li> <li>• Proprietary</li> </ul>
Data import/export only	<ul style="list-style-type: none"> <li>• Based on open standard</li> <li>• Proprietary</li> </ul>
No integration method	

**Table 10: Integration models**

### 5.2.6 Vendor Attributes

Although not directly related to the specific product, attributes of the vendor selling the software are obviously important for every buying decision and the analysis of the overall market.

Vendor Attributes	Subtypes
Primary source of capital	<ul style="list-style-type: none"> <li>• Public company (post-IPO)</li> <li>• Venture-capital financed</li> <li>• Bootstrapped (i.e. no formal source of external capital)</li> <li>• Cross-subsidized (i.e. vendor derives his revenue mainly from other line of business and uses proceeds to develop a web-based enterprise application)</li> </ul>
Age of company	
Geographic location(s)	
Product strategy/ Breadth of product palette	<ul style="list-style-type: none"> <li>• Focus on specific, but broader application</li> <li>• Offers a niche application only</li> <li>• Offers a suite of applications</li> </ul>
Service and support offerings	

**Table 11: Vendor attributes**

## 5.3 Vendor Market

### 5.3.1 Sample Analysis

#### 5.3.1.1 Application Type

An analysis of the different application types that the companies in the sample offer shows the broad variety of web-based solutions:

<i>Application Type</i>	<i>Number of offerings</i>
CRM	23
Team collaboration	23
Finance	18
Application Development	15
Calendar	15
ERP	12
Project management	12
HR	11
Content Management	9
Conferencing	8
Business Intelligence	7
Imaging	6
Salesforce management	6
Spreadsheet	6
Wiki	6
Word Processing	6
Storage	5
e-Mail	4
Portal	4
Presentation	4
Infrastructure	4
Blog	3
Database	3
Search	3
e-Learning	2
Mobile Access	2
Professional Services Automation	2
Supply Chain	2
Workflow	2
Appliance development/management	1
BPM	1
Field Service Management	1
Healthcare	1

RSS	1
Social Networking	1

**Table 12: Frequency of application types**

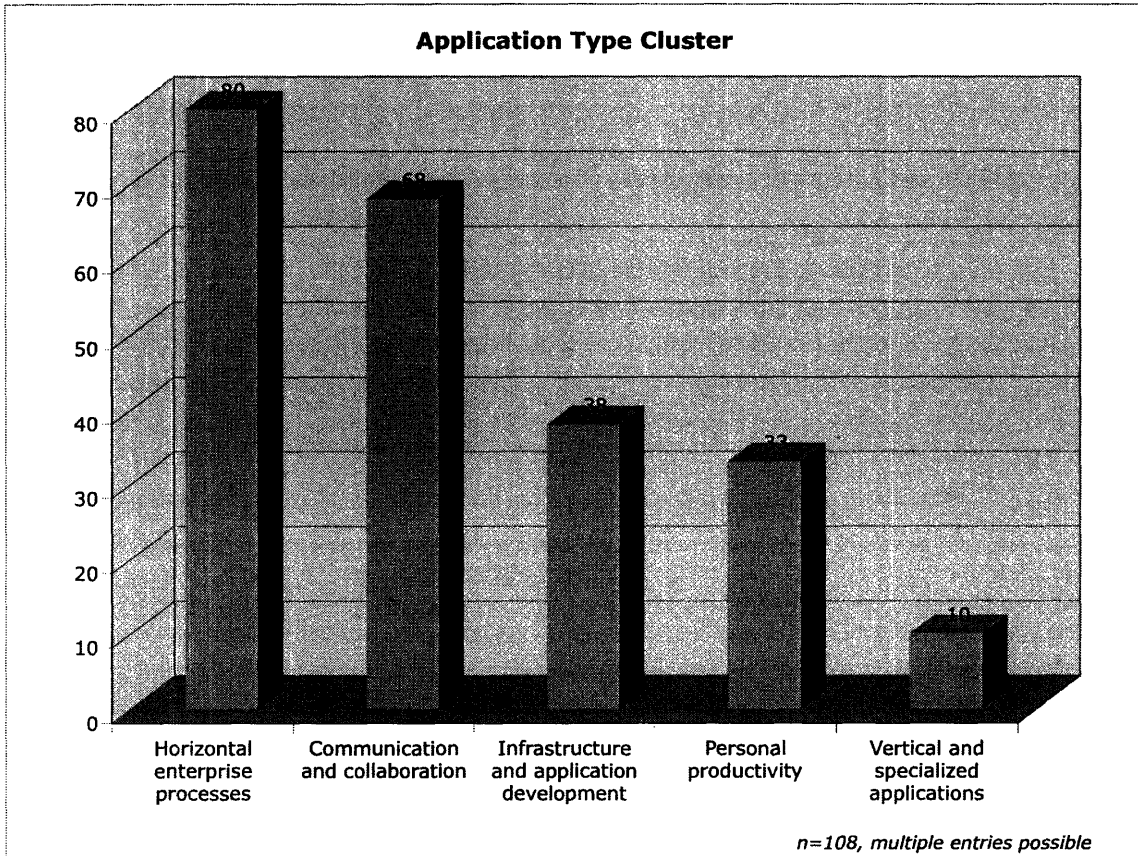
Since many companies offer multiple products, multiple entries for application type were possible.

CRM and team collaboration are the most popular application types. In both cases, there are other application types that are closely related. For CRM, these are sales force management and some vertical applications, such as field service management and professional service automation. For team collaboration, i.e. platforms that enable teams to share information, documents, etc., the related applications are project management, conferencing, wikis, and content management.

Often, vendors do not clearly differentiate between these application types in terms of functionality. For the purposes of the sample, the type of application that a vendor company declared itself was used.

Many vendors offer functionalities for application development, in almost all cases in combination with other applications. One successful example is the AppExchange platform run by Salesforce.com that enables third-party developers to write companion applications for the Salesforce.com CRM software (Salesforce.com, 2007b).

When grouped according to the structure outlined in section 3.2.2, the following clusters emerge:

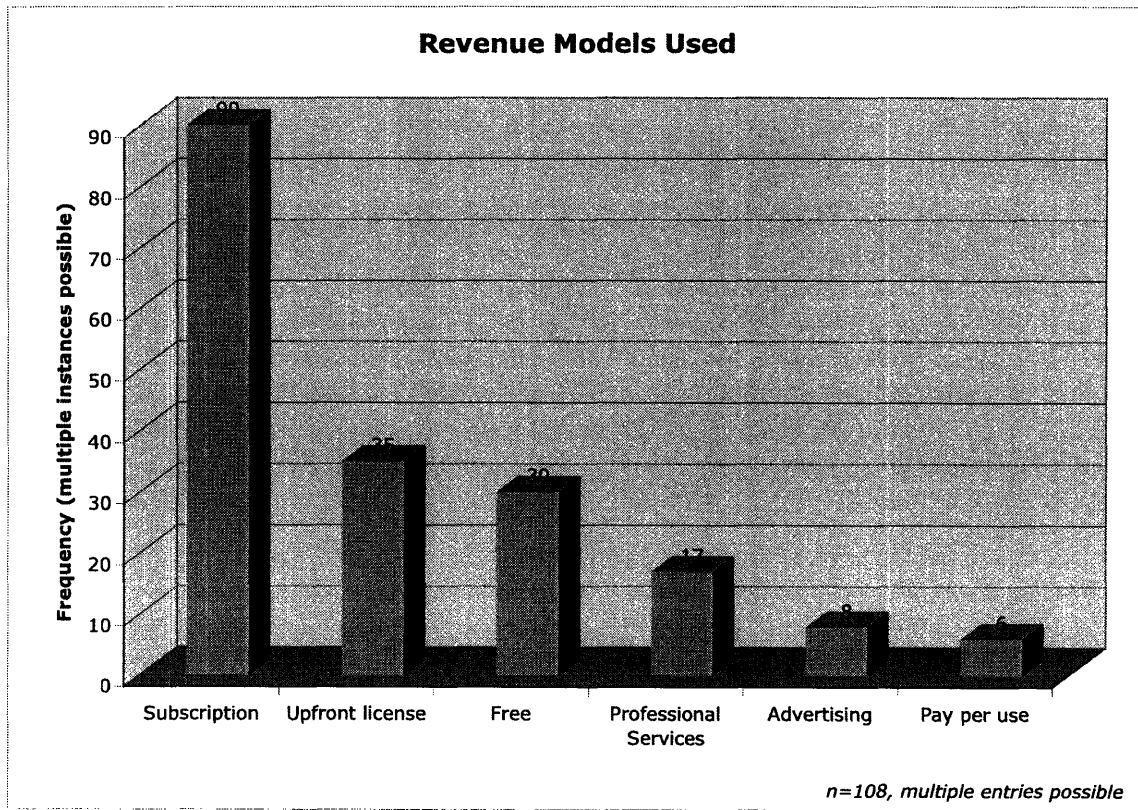


**Figure 3: Application type clusters**

The automation of horizontal enterprise processes (such as ERP, CRM, financial applications) is the most frequent use of web-based software, with communication and collaboration a close second. Infrastructure and application development are important as a technical basis, and personal productivity applications are also playing an important role.

Currently, web-based software is not quite as often used for vertical and specialized applications yet. However, there could be a bias in the sample, because very specialized applications are often difficult to find and often not sold as a stand-alone product.

### 5.3.1.2 Revenue Model



**Figure 4: Revenue models**

The revenue model describes how a vendor makes money, i.e. what the source of revenue payment streams is. The sample allowed for combinations of different revenue models.

The by far most popular model, used by 90 of 108 companies, are subscription fees (monthly or annually). This is not particularly surprising, since hosted applications are most typically combined with a SaaS business model.

35 companies also sell their software for an upfront license fee. This is often combined with alternative deployment methods. For instance, there are companies that offer their product as a hosted service based on a subscription fee, but also alternatively as a software product for local installation that is sold with a perpetual license.

30 companies, or more than a quarter, offer free versions of their products. There are two different types of free offerings:

- Some subscription-based products are available in a free version that only offers the most basic functionality. It is actually quite common for SaaS-oriented companies to offer multiple versions of a product with different functionality at different price points. Often, the simplest version is free in order to attract new customers.
- A few very young startup companies do not charge at all (yet) for their product. The motivation is probably to build a user base as quickly as possible that later

can be migrated to a subscription-based model. Some of these companies are apparently also hoping to be acquired by a bigger player (such as Google, Yahoo or Microsoft) and intend to build a large user base quickly in order to maximize the company valuation.

17 companies sell professional services, typically for integration purposes. This is a similar model to the one used by many traditional enterprise software companies that generate some of their revenue from consulting and systems integration services.

Only 8 companies use advertising as a source of revenue, most famously Google. Advertising is almost always used as an additional source of revenue in combination with other models. Quite apparently, the rise of the advertising-financed application, predicted for instance by Microsoft CTO Ray Ozzie (Ozzie, 2005), hasn't happened yet.

Finally, 6 companies offer pay-per-use models that charge users based on the number or volume of transactions carried out with the software.

The following table illustrates how many different combinations of revenue models vendors are using:

<i>Revenue Model Combination</i>	<i>No. of vendors</i>
Subscription only	33
Subscription, Upfront license	17
Free only	12
Subscription, Free	12
Subscription, Upfront license, Professional Services	7
Subscription, Professional Services	5
Subscription, Pay per use	5
Subscription, Upfront license, Free	3
Subscription, Advertising	2
Upfront license only	2
Subscription, Upfront license, Advertising	2
Advertising, Free	1
Professional Services only	1
Upfront license, Advertising	1
Upfront license, Professional Services	1
Subscription, Pay per use, Professional Services	1
Subscription, Advertising, Professional Services	1
Subscription, Upfront license, Free, Advertising	1
Subscription, Upfront license, Free, Professional Services	1
<b>Total</b>	<b>108</b>

**Table 13: Revenue model combinations**

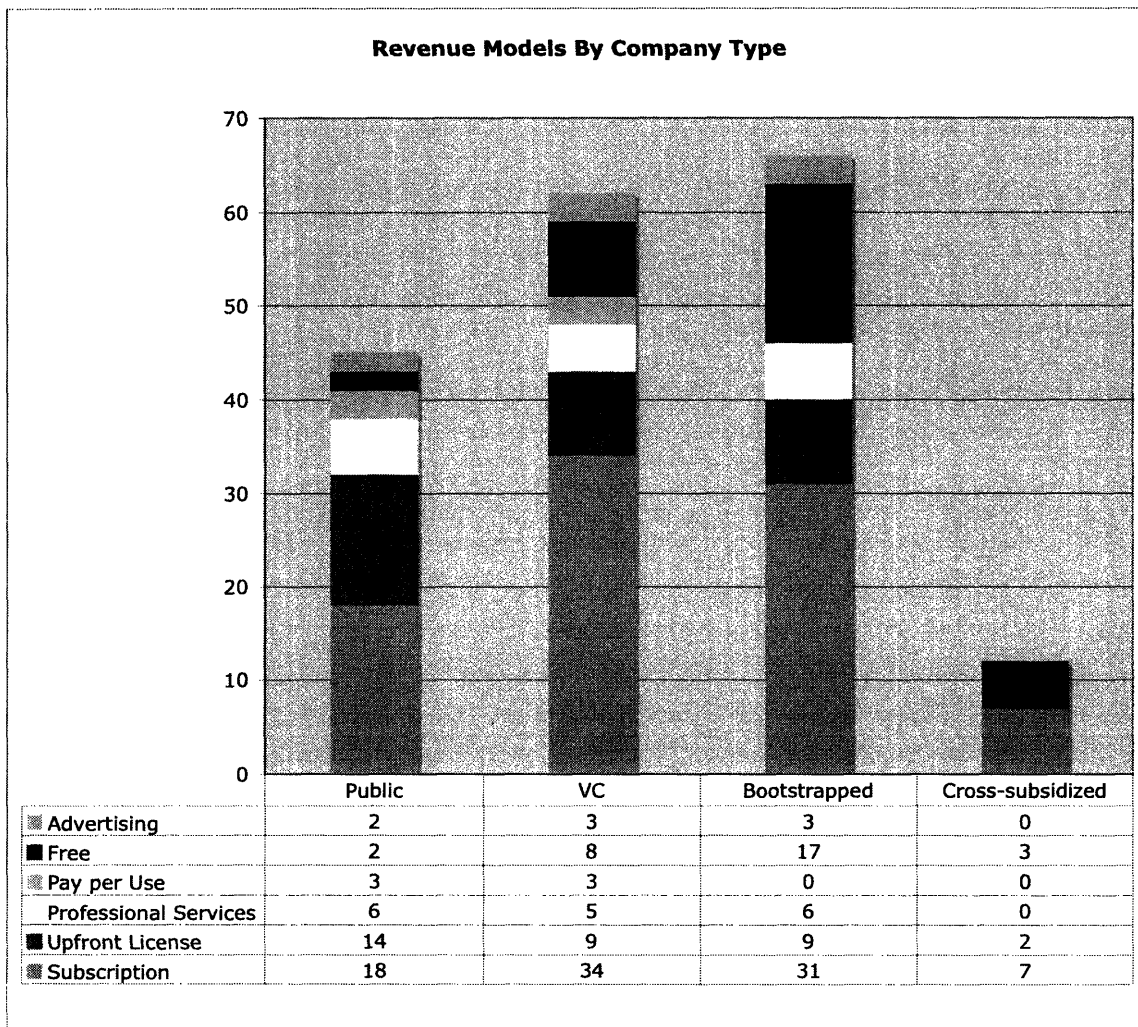


The by far strongest approach with 33 instances is to offer only a subscription model. Another 12 companies, mostly very young startups, currently only offer a fully free version (without advertising or any other revenue source).

There are another 18 combinations of revenue models. This variety of approaches shows clearly that companies currently are experimenting with a lot of different strategic approaches.

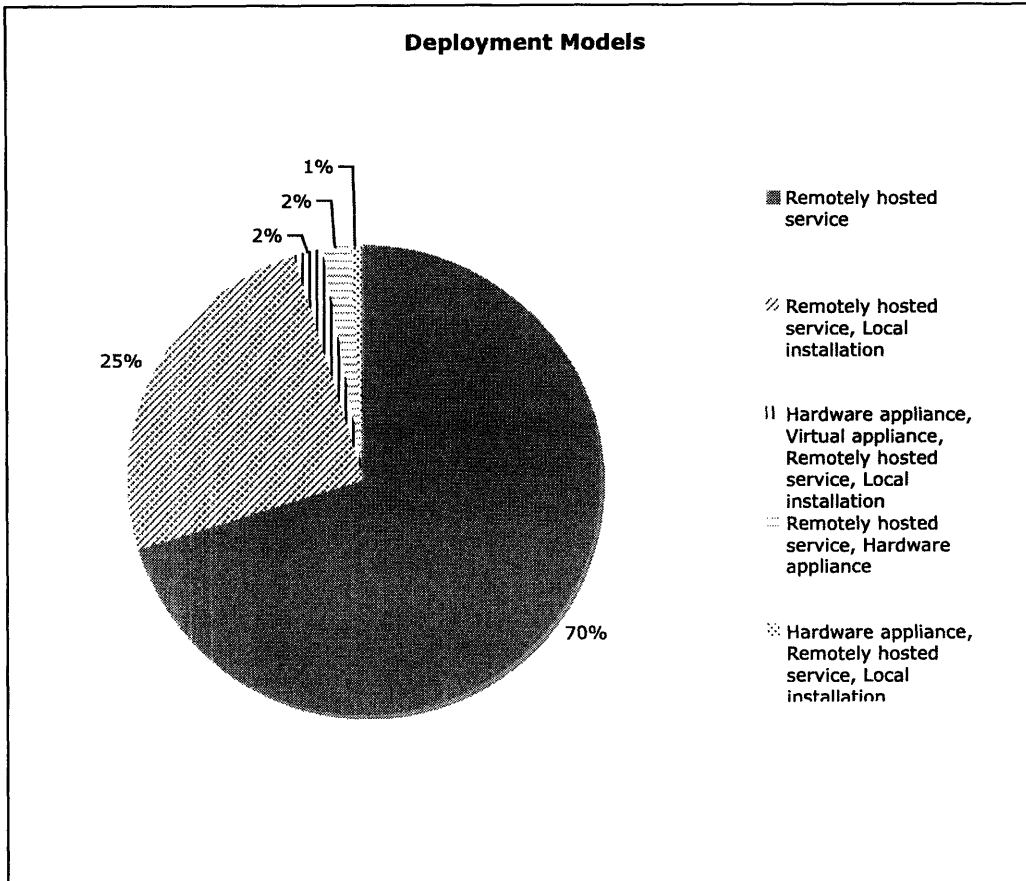
However, the subscription-based model is apparently slowly emerging as the dominant revenue model.

An analysis of the revenue models used by the different types of company (defined by the main source of capital) shows that the more mature public companies use traditional upfront licenses far more often than the VC-backed or bootstrapped startup companies. Bootstrapped companies most frequently offer free versions of their products.



**Figure 5: Revenue Models by Company Type**

### 5.3.1.3 Deployment Model



**Figure 6: Deployment model combinations**

Since the sample selected only companies offering a hosted model of deployment, obviously all companies provide at least this option. But 30% of companies also offer combinations with other models.

The most popular approach is to combine a hosted solution with an optional locally installed version. There are three subsets covered by this cluster:

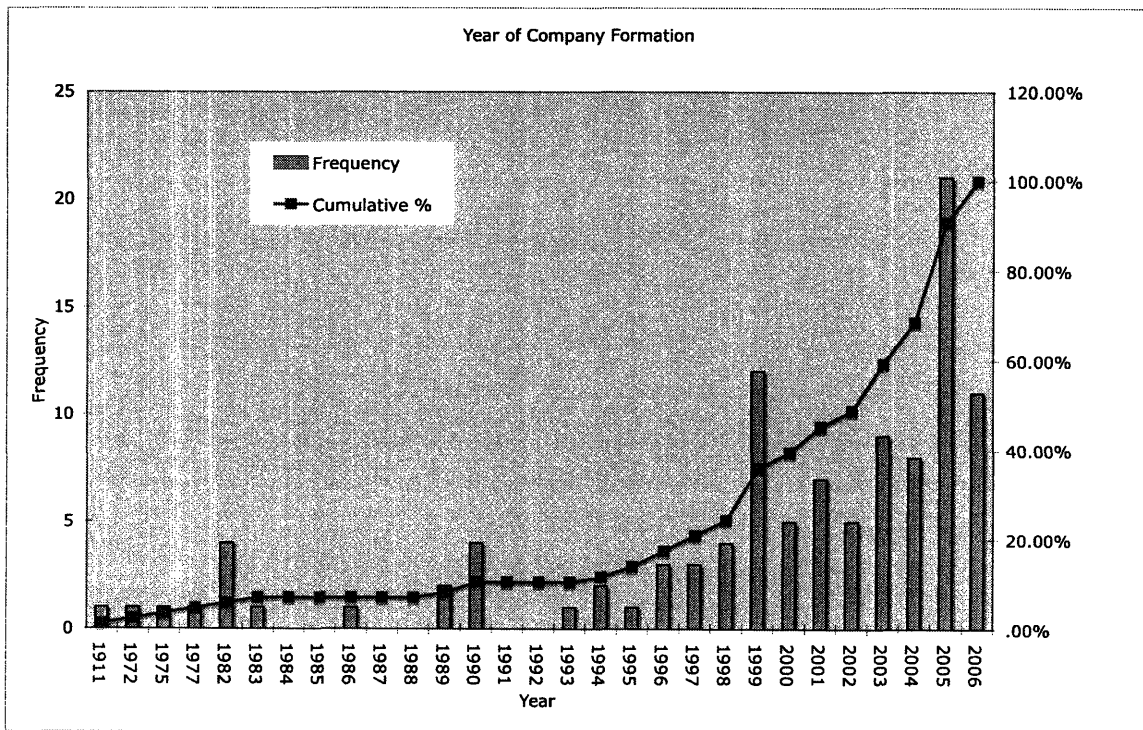
- Some products are offered with more or less identical functionality as both a hosted version or a traditional locally installed software product. This gives customers a better choice of their preferred way to use a particular product. An example is ThinkFree, a vendor that offers a Java-based office suite in both online and downloadable versions (ThinkFree, 2007).
- Some vendors offer a part of their product portfolio only as a hosted solution and another only as installed software. For example, blog vendor Blogtronix's enterprise solution is only available as a hosted solution, but there is a downloadable "lite" version for smaller customers (Blogtronix, 2007).
- In some other cases, the base product is a hosted solution, but there are optional extensions that require local installation and extend the functionality of the prod-

uct. For instance, CRM vendor RightNow Technologies offers a local version of its product as an add-on for its hosted CRM suite (RightNow, 2007). The goal is to enable traveling salespeople to use the product’s functionality while they are not online.

Two vendors, SugarCRM and Etelos, offer all four deployment options that the sample captured.

### 5.3.1.4 Firm Age

The following histogram shows the frequency of the years when companies were founded.



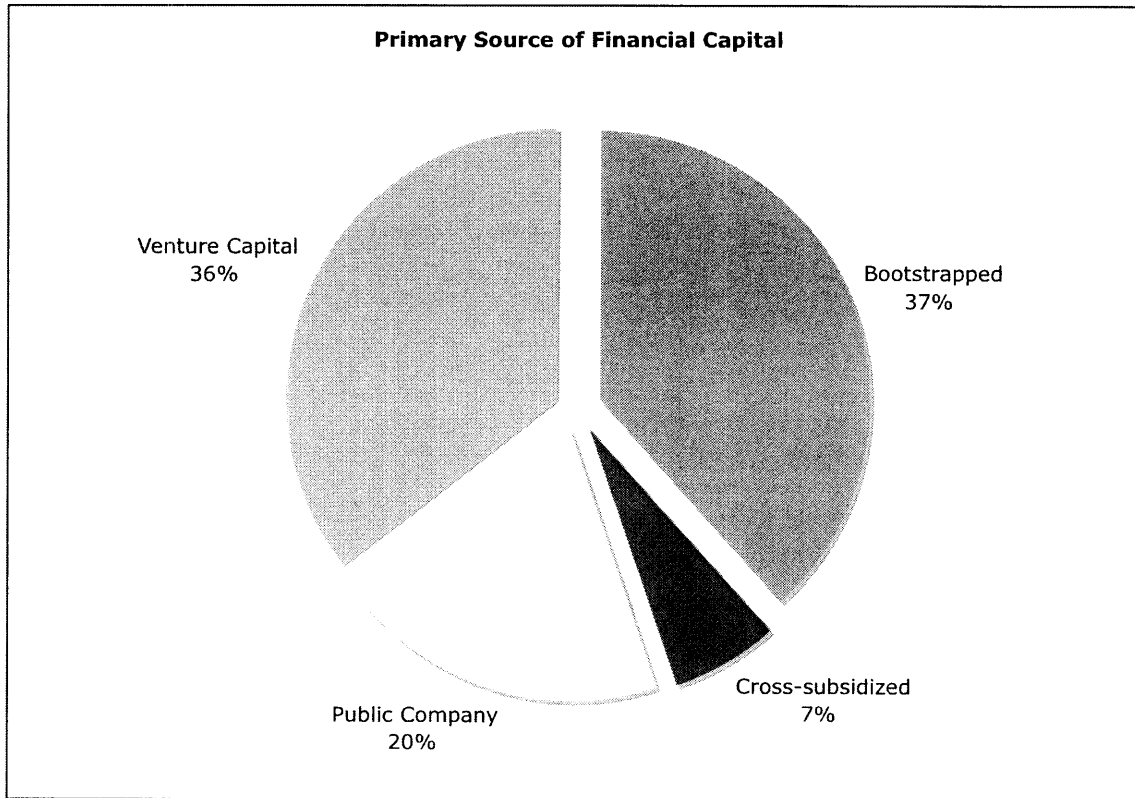
**Figure 7: Year of company formation**

The companies founded before 1995 are traditional software companies that now also offer web-based solutions as an addition to their existing product portfolio.

The most notable peaks in the frequency of founding years are 1999 and 2005, coinciding with strong investment activity in the Internet industry and IT industry in general. According to the National Venture Capital Association, 1999 and 2000 were the years with the largest venture capital investments by far over the last 10 years (NVCA, 2006), with total investments of \$53.5 billion and \$104.4 billion, respectively. However, total investment levels in 2005 and 2006 were far lower at \$22.3 billion and \$25.5 billion, but still considerably stronger than the years between 2002 and 2004.

It is furthermore significant that more than 50% of the companies were founded in or after 2002, i.e. after the end of the first “Internet bubble”. Almost 30% were founded in 2005 or 2006. This points to a high level of entrepreneurial activity in this sector.

### 5.3.1.5 Financing



**Figure 8: Primary source of financial capital**

The figure above analyzes what the main source of financial capital for the firms in the sample was at the time of the analysis.

20% of the companies are public, i.e. their shares are traded on the stock market. These are mostly the established older companies that now also offer SaaS-based solutions, such as Oracle or Microsoft. Only six younger companies that specialize in web-based software are already public: WebEx<sup>2</sup>, Salesforce.com, RightNow Technologies, Salary.com, Concur Technologies, and Taleo. Foldera, a startup that is still in the pre-revenue stage, is publicly traded as an over-the-counter penny stock.

---

<sup>2</sup> Cisco Systems announced the acquisition of WebEx for \$3.2 billion on March 13, 2007 (White and Cheng, 2007). Since the merger wasn't completed yet at the time of this writing, WebEx is treated as an independent company in this study.

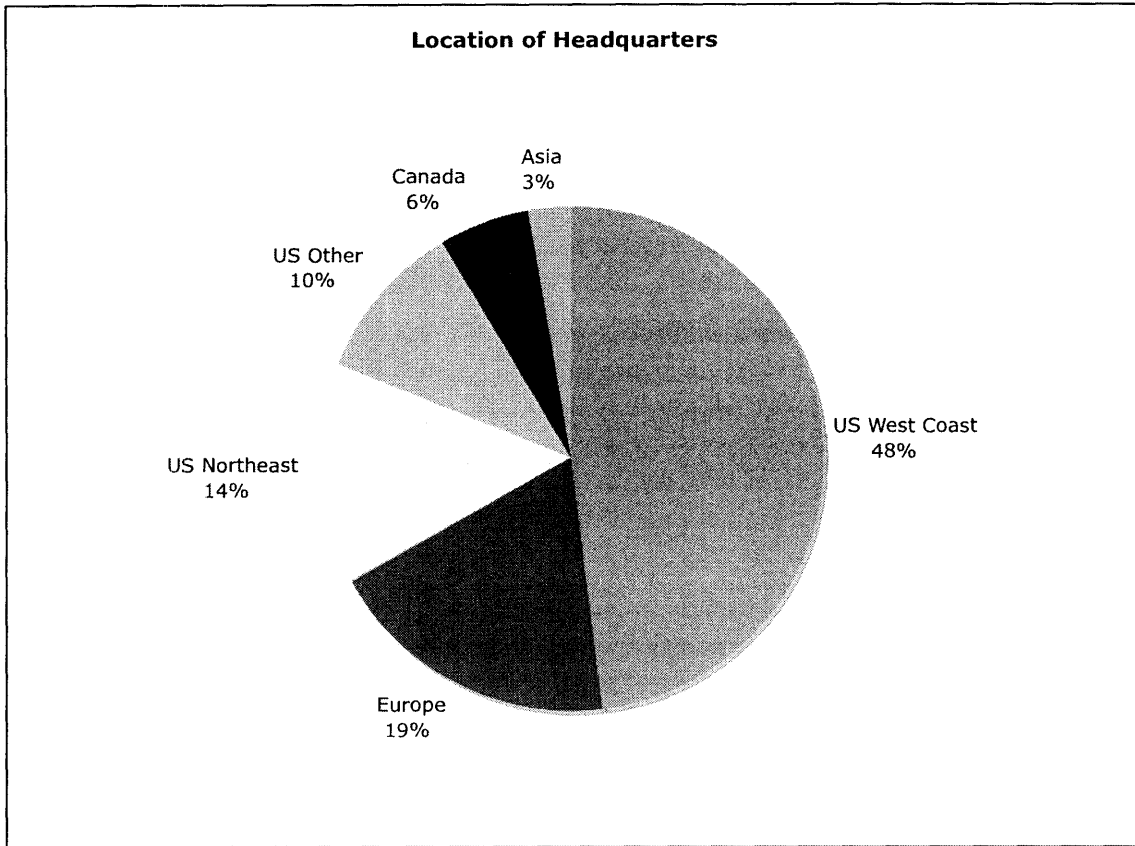
For 36% of the companies, the main source of funding is formal venture capital investment. The importance of venture capital is not surprising, since the software industry has traditionally been one of the most important fields for investments for the venture capital community.

More surprising however is the fact that more than a third of the companies are “bootstrapped”, i.e. financed by their founder’s own investment and/or business angels close to the company. In other words, these companies did not receive any formal investments from the public or private capital market. Judging from the sample, there are three types of bootstrapped companies:

- Very young companies that are still in the development phase of their first product. Since web technology is very cost-efficient, it is often possible for a small team of founders to develop a first product with very little need for capital. A big advantage of web-based software is the possibility to release a product to the public at a very early stage with very little cost. Therefore, these companies are able to offer products on the market without the need to build sales channels and other costly infrastructure.
- Some slightly older companies (founded in 1995-2004) were apparently able to finance their growth from early revenues. This is one of the most attractive characteristics of web-based software: Since there is a good chance to generate early revenues with relatively little upfront investment, companies can be built based on their own cash-flow.
- In some cases, successful entrepreneurs have founded new startups in this space. These wealthy individuals are able to finance their new companies from their own assets. A typical example is Dave Duffield, founder of PeopleSoft, who is now a co-founder of web-based ERP startup Workday (Workday, 2007).

The category “cross-subsidized” was assigned to companies whose core business is clearly in another field, typically in professional IT services. These companies are trying to build a product-based business by leveraging their in-house knowledge, but still rely on their core business for financing.

### 5.3.1.6 Geographic Aspects



**Figure 9: Location of headquarters**

More than 70% of the companies in the sample have their headquarters in the United States. By far the most important single location, with more than 40% of the total sample, is California, or more specifically, Silicon Valley. 19% of companies are headquartered in Europe, but only 3% in Asia.

There is certainly a good chance that the sample is biased towards U.S. companies, since most sources of information for the creation of the sample were of American origin, and almost all sources were in the English language (with a few in German and French). The sources for European and Asian markets were much less accessible and well structured.

However, this very fact illustrates a problem for software companies not headquartered in the United States: Not only are markets outside of the US smaller and more fragmented; the infrastructure for the promotion of a new product (e.g. trade press, conferences, trade shows) is also much less developed.

### 5.3.2 Conclusions from Sample Analysis

The following main conclusions can be drawn from the analysis above:

- Web-based enterprise software is used for a variety of application types, but horizontal enterprise processes and collaborative applications are the dominant clusters.
- There are still many different revenue model combinations being used by vendors. However, the subscription model is clearly the most frequently used variation.
- About 30% of vendors are using other deployment models in combination with hosted solutions, often to enhance functionality or increase customer choice.
- Web-based software is a clearly “hot” sector with considerable entrepreneurial activity and many new companies that were founded over the last three years.
- Bootstrapping a software company is becoming more feasible due to low investment needs, and therefore this type of funding is the most important in this sample. However, venture capital is still an almost equally important source of funds for this type of company.
- The market is clearly dominated by American companies, predominantly those located in Silicon Valley.

### 5.4 Case Studies

The following sections profile a selection of companies that are currently active in the market for web-based enterprise software. This selection represents typical company types and therefore can be useful to understand some of the most frequently observed strategic patterns:

- **Salesforce.com and RightNow Technologies:**  
Public, specialized SaaS companies, focused on one application type.
- **NetSuite:**  
Privately held, specialized SaaS company that offers a broad suite of business applications.
- **Xactly and Smartsheet:**  
Recently founded SaaS startups with a clear niche strategy.
- **37signals and Zoho/AdventNet:**  
Privately held IT companies, originally active in other segments, that sell web-based product suites.
- **Google and Microsoft:**  
Large, established companies that offer web-based software as an extension of their core business.
- **SAP and Oracle:**  
Established business application companies that are starting to offer SaaS solutions as a part of their portfolio.

### 5.4.1 Salesforce.com

Salesforce.com, based in Redwood City, California, was founded in 1999 (Salesforce.com, 2007a) and is currently the largest public software company that is fully focused on Software as a Service<sup>3</sup>. CEO and Chairman Marc Benioff spent 13 years at Oracle Corporation before founding Salesforce.com. The company went public in the summer of 2004, and its current market capitalization is \$4.92 billion.

According to its website, Salesforce.com had 29,800 customers with a total of 646,000 subscribers at the beginning of 2007. It had revenues of \$497.1 million for the year ending January 31, 2007, growing from \$308.9 million in the previous year.

The company's main application focuses on customer relationship management and sales force automation. Originally, Salesforce.com's main target customers were smaller companies, but it now boasts several large customers, such as AMD, Kaiser Permanente, and Cisco (Salesforce.com, 2007a). Salesforce.com users are charged a monthly fee per user, starting at a list price of \$65/month.

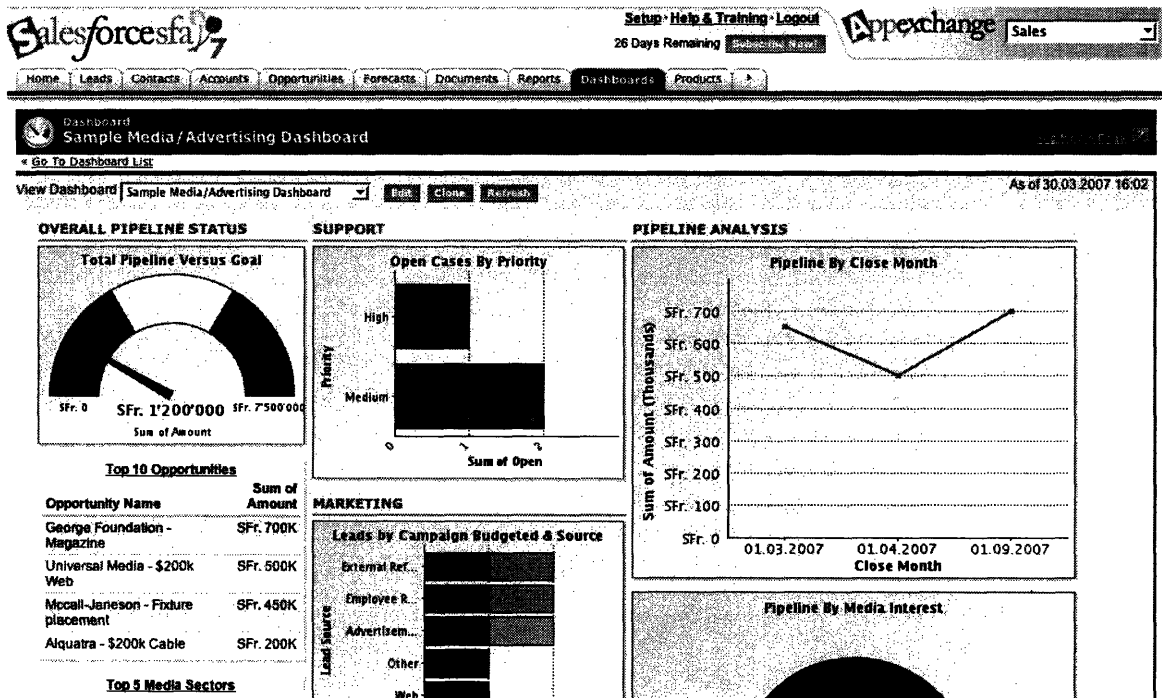


Figure 10: Salesforce.com user interface

In 2005, Salesforce.com introduced its AppExchange platform that enables third-party developers to write applications that connect to Salesforce.com's system. Developers can sell their applications on the AppExchange marketplace to existing Salesforce.com cus-

<sup>3</sup> Measured by market capitalization as of March 30, 2007, according to Yahoo Finance.



tomers. According to the company, there are currently over 500 applications available, covering a wide range from document management to integration tools (Salesforce.com, 2007b).

With Apex, Salesforce.com in late 2006 introduced a programming language for on-demand application modules that can run on the company's platform. This language allows outside developers to extend and customize Salesforce.com's core application, including changes to the basic data model. It is even possible to create entirely new applications that integrate closely with the core CRM application. This new functionality is apparently Salesforce.com's reaction to customers' concerns about the lack in flexibility of a hosted application. At the time of this writing, Apex is available as a beta version.

Furthermore, the company offers a platform called Successforce.com that enables its customers to exchange best practices and tips. Salesforce.com even invites customers to discuss strategic issues such as "Should Salesforce re-brand its product family?"

Quite obviously, Salesforce.com is trying to build an ecosystem of third-party developers for its platform. Traditional software companies have done this for decades, and it is certainly a major reason for the success of Microsoft, Oracle and others (Gawer and Cusumano, 2002). Judging from the sample examined in section 5.2, Salesforce.com is currently by far the most successful SaaS-focused company doing this kind of platform building. The company is trying to build a two-sided software platform (Evans, Haigu and Schmalensee, 2006) by enabling direct interactions between application users and vendors of new programs that build on AppExchange.

The company's revenue growth rate was around 80% per year for its first few years, but has slowed down to 60% for the 2007 fiscal year (ending in January 2007). The company was profitable between FY2004 and FY2006, but incurred an operating loss in FY 2007.

Figure 11 shows Salesforce.com's income statement in more detail. One interesting fact is that the company generates more than 90% of its sales from subscription and support fees. Only 9% are from professional services, which are currently not profitable (Salesforce.com, 2006).

Salesforce.com's R&D expenses are comparatively small with only 9% of revenues (compared to a software industry average closer to 17%; Vincent et al., 2005). This might be a result of its specific model. Vendors of hosted applications do not have to maintain many different versions of their products the way traditional software companies have to. As a consequence, it is plausible that the total R&D costs are lower, although Salesforce.com has introduced several remarkable innovations over the last few years.

The company spends a lot, a full 51% of revenues, on marketing and sales. This is understandable for a relatively young company with a new approach that is still not known to many potential customers. Salesforce.com has only managed to decrease this cost factor slightly over the last few years, and as a result, its operating margin is still small. Operating costs without R&D account for a full 68% of revenues. Traditional software companies of a comparable size, such as Lawson Software or i2 Technologies, spend considerably less on Sales, Marketing, and G&A (Lawson Software, 2006; I2, 2006).

The line item "other income-affecting positions" summarizes income components that are not directly related to operations. For Salesforce.com, these are primarily interest income

(\$14.9 million in 2006), provisions for taxes (-\$9.8 million in 2006), and provisions for a consolidated joint venture in Japan (-\$2.2 million).

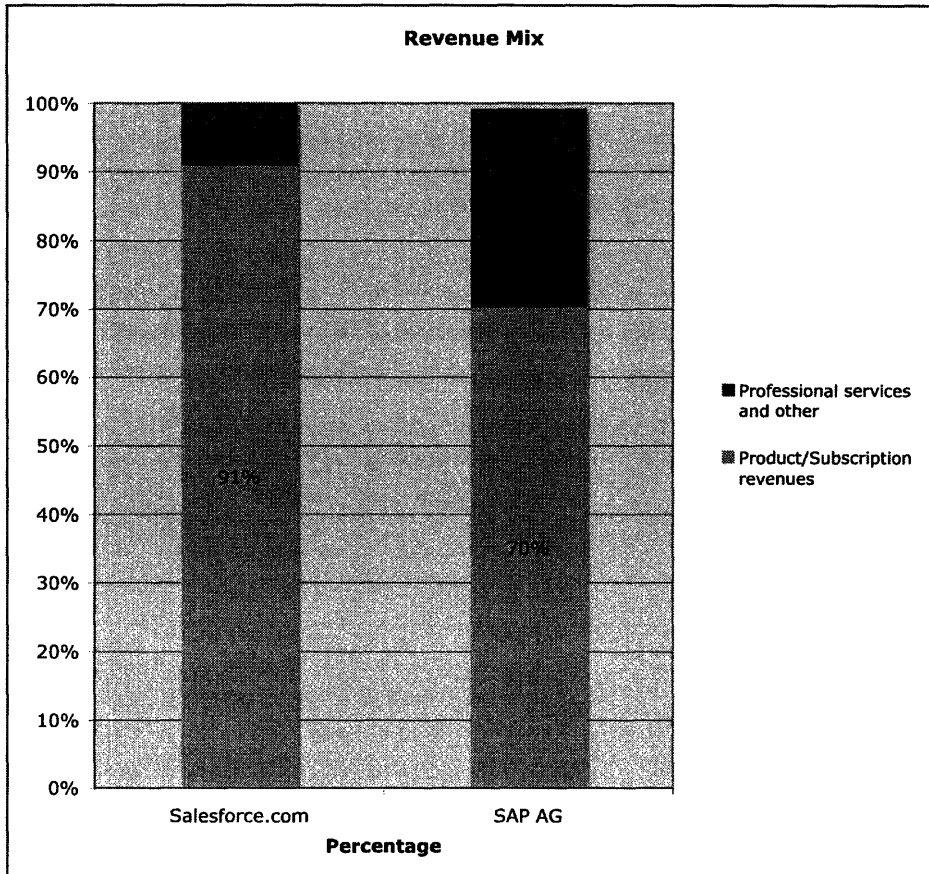
**Salesforce.com Income Statement**  
(in USD 1,000)

Fiscal Year	2007	2006	2005	2004	2003
<b>Revenues:</b>					
Subscription and support	451,660	280,639	157,977	85,796	47,656
<i>Percentage of total</i>	91%	91%	90%	89%	93%
Professional services and other	45,438	29,218	18,398	10,227	3,335
<i>Percentage of total</i>	9%	9%	10%	11%	7%
<b>Total revenues</b>	<b>497,098</b>	<b>309,857</b>	<b>176,375</b>	<b>96,023</b>	<b>50,991</b>
<i>Revenue growth comp. to previous year</i>	60%	76%	84%	88%	
<b>Cost of revenues:</b>					
Subscription and support	61,457	34,457	12,727	7,782	7,199
<i>Gross profit for subscr. and support</i>	390,203	246,182	145,250	78,014	40,457
Professional services and other	57,433	34,669	20,727	9,491	3,164
<i>Gross profit for professional serv.</i>	-11,995	-5,451	-2,329	736	171
<b>Total cost of revenues</b>	<b>118,890</b>	<b>69,126</b>	<b>33,454</b>	<b>17,273</b>	<b>10,363</b>
<b>Gross profit</b>	<b>378,208</b>	<b>240,731</b>	<b>142,921</b>	<b>78,750</b>	<b>40,628</b>
<i>Percent of Revenues</i>	76%	78%	81%	82%	80%
<b>Operating expenses:</b>					
Research and development	44,614	23,330	9,822	6,962	4,648
<i>in percent of revenues</i>	9%	8%	6%	7%	9%
Marketing and sales	252,935	149,598	96,311	54,600	33,522
<i>in percent of revenues</i>	51%	48%	55%	57%	66%
General and administrative	84,257	47,986	30,268	16,915	12,958
<i>in percent of revenues</i>	17%	15%	17%	18%	25%
Other		-285		-3445	
<b>Total operating expenses</b>	<b>381,806</b>	<b>220,629</b>	<b>136,401</b>	<b>75,032</b>	<b>51,128</b>
<i>in percent of revenues</i>	77%	71%	77%	78%	100%
Income (loss) from operations	-3598	20,102	6,520	3,718	-10,500
Other income-affecting positions	4079	8372	826	-204	784
<b>Net income (loss)</b>	<b>481</b>	<b>28,474</b>	<b>7,346</b>	<b>3,514</b>	<b>-9,716</b>

**Figure 11: Salesforce.com Income Statement (Salesforce.com, 2006)**

A comparison to ERP market leader SAP's income statement ratios shows some of the properties of Salesforce.com's specific model and state of corporate development.

SAP, not untypical for a mature business application company, has a comparatively large percentage of consulting revenues (30% of total revenues in 2006; SAP, 2006). As already discussed, Salesforce.com derives 91% of revenues from subscription fees and support. The 9% coming from professional services seem to be a kind of loss leader, while SAP's professional services are profitable with a gross margin of 23%.



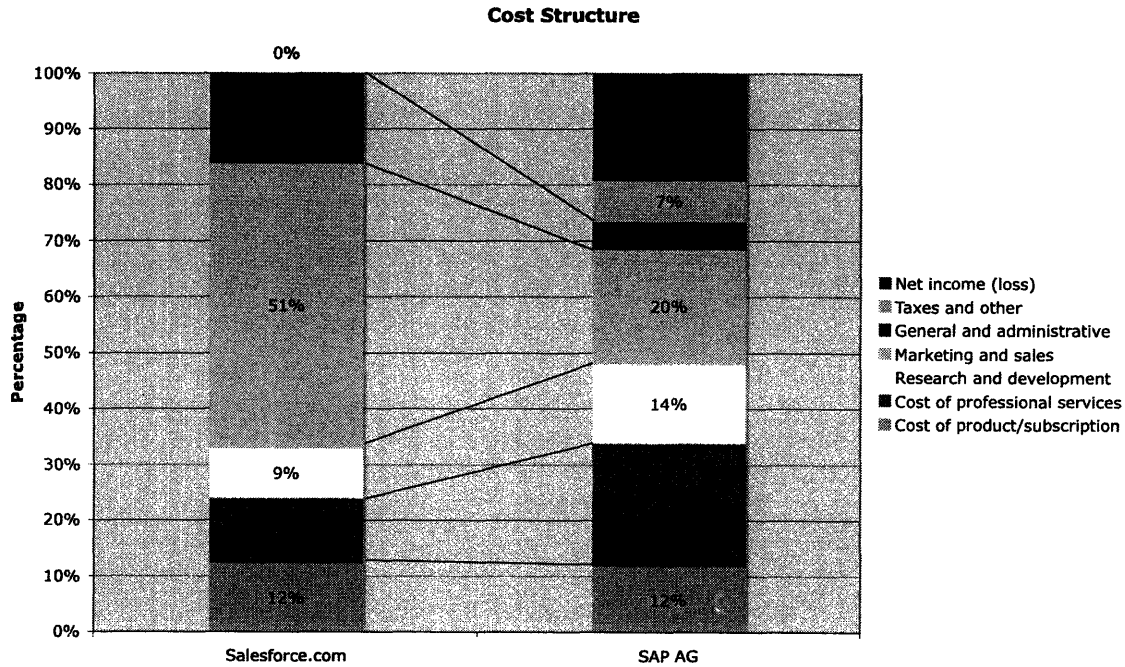
**Figure 12: Revenue mix for Salesforce.com and SAP AG**

A comparison of cost structures shows some additional interesting aspects (see Figure 13). Both companies spend a comparable percentage of revenues (12%) for the costs of its products/subscriptions. SAP's higher share of professional services leads to a larger cost block for its consulting operations. However, Salesforce.com has lower R&D costs with only 9% of revenues vs. SAP's 14%. As already mentioned, the by far biggest cost block for Salesforce.com are its sales and marketing expenses, which account for 51% of revenues vs. SAP's 20%. Finally, Salesforce.com spends 17% of its revenues on General & Administrative, SAP only 5%.

The resulting net income for Salesforce.com is almost 0%, while SAP earns a net income margin of 20%.

Obviously, comparing a mature company such as SAP with a startup such as Salesforce.com has its limitations. Particularly the lower marketing and G&A costs for SAP are certainly partly due to its established processes and strong market position. Typically, as a software company grows and matures, these cost items do not scale with the same ratio as revenues. If Salesforce.com can scale its revenue base without increasing its G&A and marketing costs proportionally, it will be in a position to build a very profitable business, given its high gross margin.

Furthermore, it is remarkable how low Salesforce.com’s costs for its core product development and service provisioning are. This might be partly due to the particular characteristics of the SaaS model. However, Salesforce.com in its short history so far was not forced to release a completely new product generation, such as for instance SAP’s shift from SAP R/2 to R/3. Building a completely new product while still servicing an older version for existing could potentially increase R&D and support costs in the future.



**Figure 13: Cost structures of Salesforce.com and SAP AG**

#### 5.4.2 RightNow Technologies

RightNow Technologies was founded in 1997 and is headquartered – somewhat unusually for a high-tech company – in Bozeman, Montana. The company offers a hosted CRM solution, similar to Salesforce.com’s products. RightNow went public in the summer of 2004 (briefly after Salesforce.com) and currently has a market capitalization of \$548m (according to Google Finance, April 3, 2007).

According to the company’s website, it currently has approximately 1,800 customers around the world (RightNow. 2007), including some large corporations such as British Telecom, Nikon and Continental Tire.

RightNow focuses on its actual core application and does not offer a development platform comparable to Salesforce.com’s AppExchange.

The company’s income statement shows a larger percentage of professional service revenues (22% in 2006) compared to Salesforce.com. Its total gross margin of 71% is low compared to pure software product companies, but still higher than that of many mixed-

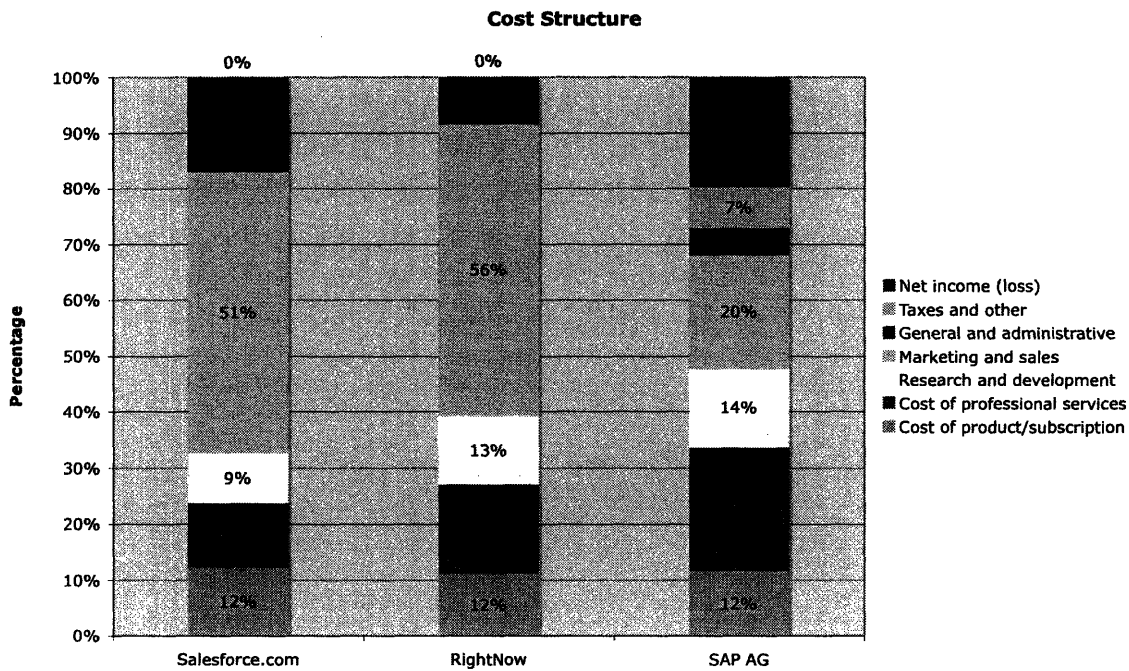
model companies. SAP for instance had a gross margin of 66% in 2006. However, RightNow last year incurred a significant loss after two profitable years.

**RightNow Technologies Income Statement**  
(in USD 1,000)

	Fiscal Year	2006	2005	2004
<b>Revenues:</b>				
Subscription and support		86,257	67,944	49,764
<i>Percentage of total</i>		78%	78%	
Professional services and other		24,131	19,204	12,000
<i>Percentage of total</i>		22%	22%	
<b>Total revenues</b>		<b>110,388</b>	<b>87,148</b>	<b>61,764</b>
<i>Revenue growth comp. to previous year</i>		27%	41%	
<b>Cost of revenues:</b>				
Subscription and support		13,260	9,111	6,741
<i>Gross profit for subscr. and support</i>		72,997	58,833	43,023
Professional services and other		19,110	11,956	7,206
<i>Gross profit for professional serv.</i>		5,021	7,248	4,794
<b>Total cost of revenues</b>		<b>32,370</b>	<b>21,067</b>	<b>13,947</b>
<b>Gross profit</b>		<b>78,018</b>	<b>66,081</b>	<b>47,817</b>
<i>Percent of Revenues</i>		71%	76%	77%
<b>Operating expenses:</b>				
Research and development		14,478	10,428	7,807
<i>in percent of revenues</i>		13%	12%	13%
Marketing and sales		61,504	42,683	31,986
<i>in percent of revenues</i>		56%	49%	52%
General and administrative		9,578	6,445	4,621
<i>in percent of revenues</i>		9%	7%	7%
Other				
<b>Total operating expenses</b>		<b>85,560</b>	<b>59,556</b>	<b>44,414</b>
<i>in percent of revenues</i>		78%	68%	72%
Income (loss) from operations		-7542	6,525	3,549
Other income-affecting positions		2534	1168	-100
<b>Net income (loss)</b>		<b>-5008</b>	<b>7,693</b>	<b>3,449</b>

**Figure 14: Income Statement for RightNow Technologies**

A further comparison to RightNow's main competitor Salesforce.com exhibits a somewhat similar cost structure with some important differences (see Figure 15).



**Figure 15: Cost structures for Salesforce.com, RightNow Technologies and SAP AG**

RightNow has a larger percentage of professional service revenues, which translates into a bigger cost block for consulting operations. R&D costs are also higher and more in line with traditional software companies. This could be attributed to the fact that RightNow sells some locally installed software components, which are more difficult to maintain than Salesforce.com’s purely hosted applications.

The biggest cost block for RightNow, in a similar way as for its main competitor, is sales and marketing with 56% of revenues. The company has, however, lower G&A costs, maybe partly due to its more cost-effective location.

### 5.4.3 NetSuite

NetSuite is a vendor of web-based ERP and CRM software, founded in 1998. It is based in San Mateo, California, and is privately held. Its majority owner is Oracle CEO Larry Ellison who was also one of Salesforce.com’s first investors (Turner, 2005). The company doesn’t disclose financial information. However, according to its filing for the “Inc. 500” competition, its revenues in 2004 were \$41m (Inc. magazine, 2004), and according to other reports, its 2006 revenues were around \$70m (Lacy, 2006).

NetSuite’s products are targeted at small and medium-sized businesses and cover a wide range of ERP, CRM, accounting and eCommerce functionality. While Salesforce.com and RightNow concentrate on CRM, NetSuite sells a full business application suite that in terms of features rivals traditional ERP packages for the mid-market. Given this com-

plexity, NetSuite offers professional services and works with a number of consulting partners that customize the software for clients.

NetSuite's pricing starts at \$99/month per user for the Small Business edition but can go significantly higher than that for advanced functionality (Turner, 2005).

With SuiteFlex, NetSuite provides a technology platform for the customization of its suite, similar to Salesforce.com's Apex. Developers can also create applications that run on top of NetSuite's infrastructure with a programming language called SuiteScript.

#### **5.4.4 Xactly**

Xactly was founded in 2005 and is based in San Jose, California. The company received a first round of venture capital (\$4m) in its founding year and a second round (\$8m) in 2006 (Xactly, 2007).

The company's product, "Incent" is an on-demand sales force compensation management software. It helps customers to define, calculate and manage incentives and compensation schemes for salespeople. The included reporting functionalities provide a transparent view of sales compensations for executives, sales managers and salespeople.

The product is also available as "Incent for AppExchange", integrating directly with Salesforce.com's CRM application. Furthermore, the product offers web-service-based programming interfaces that enable the integration with existing systems.

Xactly positions its product as a companion application to popular ERP and CRM packages, including Salesforce.com, RightNow and Microsoft CRM. The company doesn't disclose revenue or customer figures, so it's difficult to assess if this niche strategy is successful.

#### **5.4.5 Smartsheet.com**

Smartsheet.com is based in Kirkland, Washington, and was founded in 2005 by former executives of CRM company Onyx Software. It hasn't received any venture capital so far, but is financed by angel investors (Cook, 2006).

The company's product is a web-based project management solution for distributed teams. It uses a spreadsheet metaphor, since, according to the company's chairman Brent Frei, most people use Microsoft Excel to keep track of projects, but complain about Excel's lack of collaboration features (Cook, 2006).

Smartsheet offers a number of templates for common management situations, such as product launches, recruiting, or financial audits. The product enables project managers to assign tasks to people, keep track of deadlines and milestones, and create status reports. Smartsheet is free in a basic version, and there are advanced subscription-based versions with prices ranging from \$25 to \$149 per month.

With its focus on project management, Smartsheet competes with a large group of both desktop-based (e.g. Microsoft Project) and web-based (e.g. 37signals) competitors. The company's strategy is apparently to offer a product that is far simpler to use than desktop-based packages, but more structured than most web-based solutions.

## 5.4.6 37signals

37signals was founded in 1999 and is headquartered in Chicago. The company started as a web design firm, but in 2004 began to develop its own hosted software products, on which it now focuses its business.

Currently, 37signals offers four products that cover project management, CRM, collaboration and online information management. According to its website, the company has 1,000,000 registered users, but this figure doesn't translate directly into paying customers, since basic versions of the company's products are available for free.

37signal's strategic focus is to offer extremely simple web-based applications with a very high level of usability. Its particular philosophy has earned the company a lot of press attention.

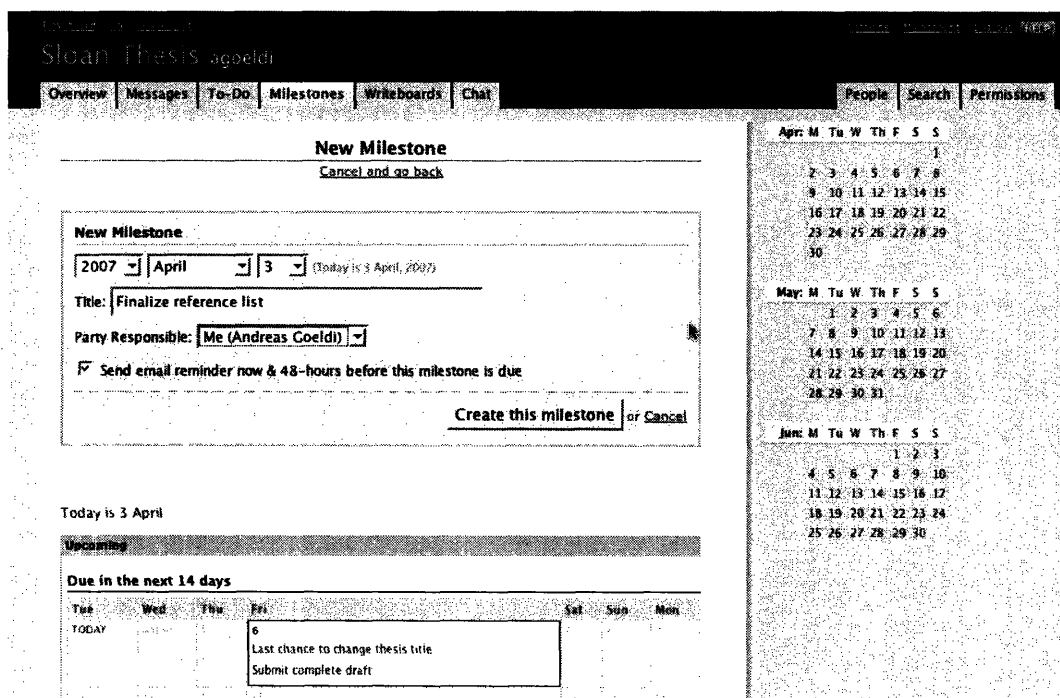


Figure 16: 37signals Basecamp project management application

Furthermore, the company is very highly regarded in the Open Source community, since it developed and published the popular Ruby on Rails programming framework (Da Silva, 2006).

The price structure for 37signal's products is straightforward and transparent: A basic version is free, and there are several subscription-based product levels, ranging from \$12/month to \$149/month per user. The various product levels are differentiated by feature sets and available amounts of online storage.

37signal's first development phase was fully financed by its founders, which was possible due to the limited size of the company. Today, the company still has only 8 employees. In 2006, 37signals received its first outside investment from Bezos Expeditions,



Amazon.com founder Jeff Bezos' investment firm (Hof, 2006). The company is still privately held and doesn't disclose financial information.

### 5.4.7 Zoho / AdventNet

The Zoho Office Suite is a product of Pleasanton, CA based IT firm AdventNet. The company was founded in 1996 and offers a wide range of IT management, security, database and testing tools. AdventNet is privately held and doesn't disclose financial information (AdventNet, 2007).

Zoho is a suite of productivity and collaboration applications including a word processor, a spreadsheet, a presentation program, a wiki, a CRM solution and a number of other tools. Currently, the Zoho suite consists of 12 different web-based applications, most of which are available for free. Only the CRM, project management and mail applications are sold based on a subscription model. Depending on configuration and storage space, users are charged \$5 - \$80 per month and application.

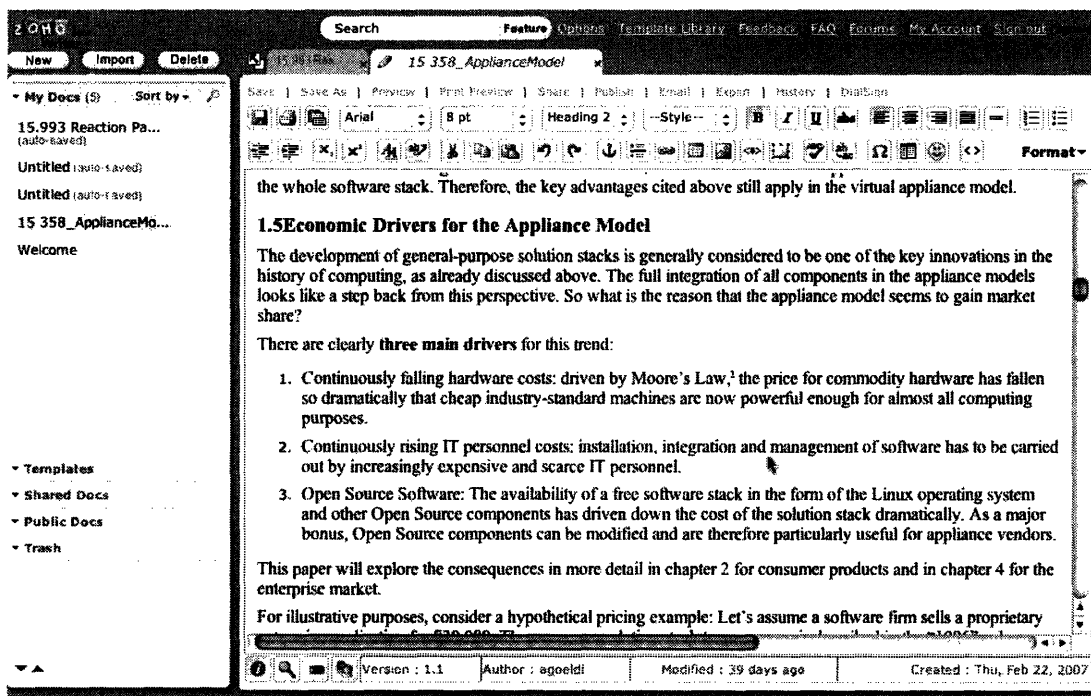


Figure 17: Zoho online word processor

Zoho also contains a simple web-based application development tool called Zoho Creator that enables users to build simple forms and databases online.

Compared to other online application suites, such as Google Apps or Microsoft Office Live, Zoho currently has probably the most complete feature set. However, the applications work to the most part independently of each other, and data sharing between pro-

grams is very limited. Like many web-based applications, Zoho products contain a number of functionalities that enable the sharing of documents with other users.

AdventNet's strategy is apparently to replicate some of the user experience of established productivity suites, such as Microsoft Office, in browser-based applications. This approach has its technical limitations, but AdventNet tries to work around these hurdles by offering some interesting tools that integrate the local desktop environment with hosted applications. For instance, a downloadable plug-in enables users to directly save Microsoft Word documents to the Zoho server and then access these documents from any web browser.

#### **5.4.8 Microsoft Office Live**

For a profile of Microsoft, please refer to section 3.2.3.1.

Microsoft in 2005 acknowledged a threat to its existing business model in an internal memo by its CTO, Ray Ozzie, titled "The Internet Services Disruption" (Ozzie, 2005). In this text that was published widely, Ozzie stated:

"The ubiquity of broadband and wireless networking has changed the nature of how people interact, and they're increasingly drawn toward the simplicity of services and service-enabled software that 'just works'. Businesses are increasingly considering what services-based economics of scale might do to help them reduce infrastructure costs or deploy solutions as-needed and on subscription basis." (Ozzie, 2005)

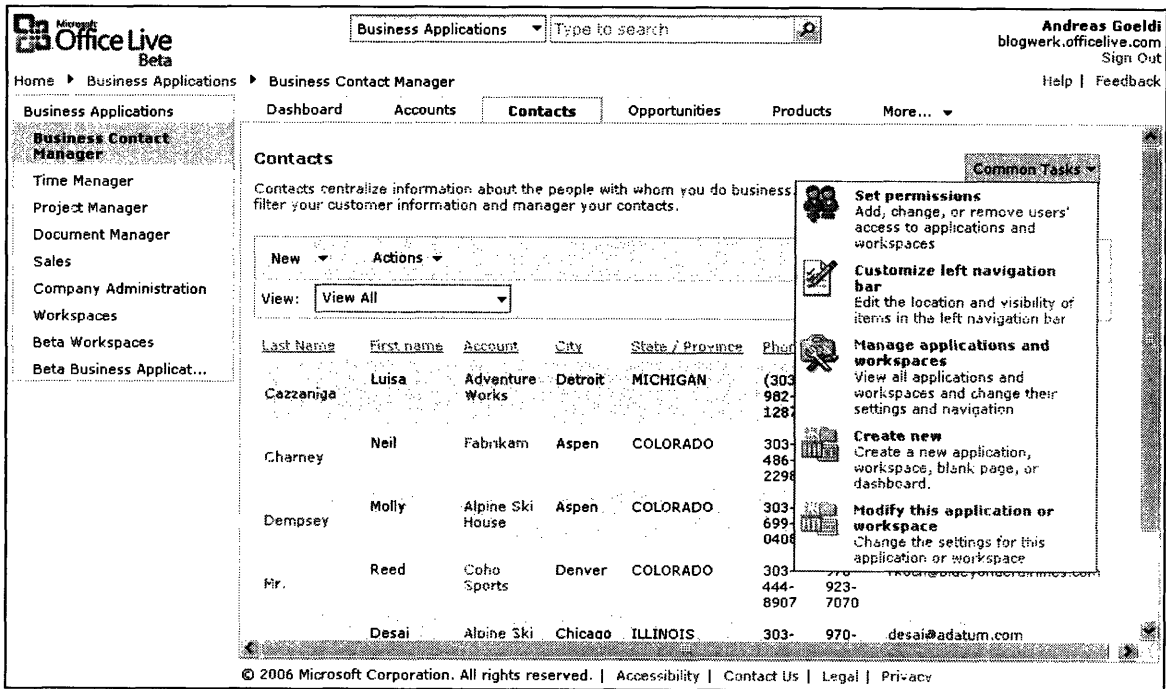
Ozzie went on to reflect critically on Microsoft's position in this new era and on the necessity for the company to proactively develop new service-based products. Many analysts agreed that web-based applications could be the biggest challenge in Microsoft's history (Economist, 2006).

A few months later, Microsoft announced a range of new web-based products under the "Live" umbrella brand (Microsoft, 2005). "Windows Live" is a collection of consumer-oriented services covering search, messaging, user generated content (such as blogs) and mapping.

"Office Live" targets small businesses and offers web hosting, simple project management tools, CRM functionalities, messaging and calendaring. Office Live currently works only with Internet Explorer 6.0 and higher, which restricts usage to Windows PCs.

The basic version of Office Live is free, but advanced versions are sold on a subscription basis for \$20-\$40/month per user. Microsoft doesn't disclose revenue or user figures for these new services.

Contrary to its name, Office Live does not offer any functionalities resembling Microsoft's Office suite, such as word processing or spreadsheets. Microsoft is positioning Office Live as a companion product, not as a replacement. Ozzie in 2006 explicitly denied any plans to create a web-based version of Microsoft Office (LaMonica, 2006).

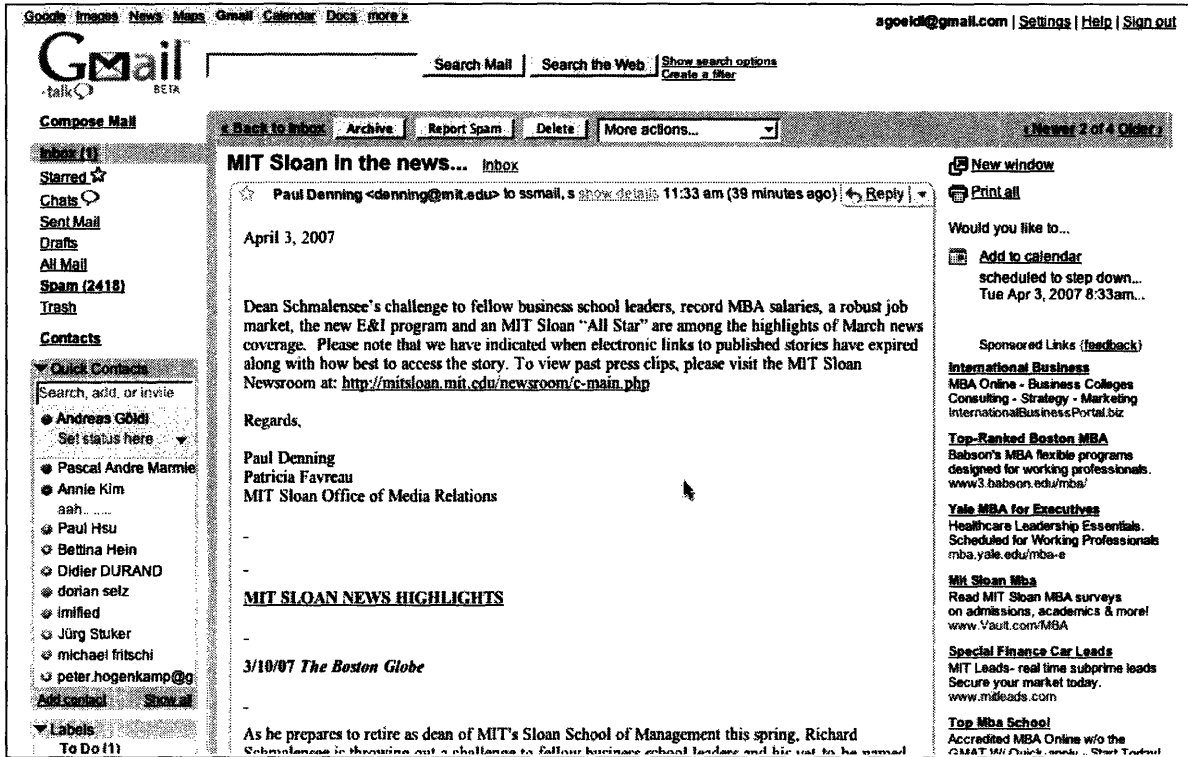


**Figure 18: Microsoft Office Live CRM application**

### 5.4.9 Google Apps

Google, the market leader in web search engines, was founded in 1999 and had its IPO in 2004. The company's current market valuation is \$144.17 billion. More than 99% of Google's revenues are generated in online advertising, but the company also has an enterprise division with revenues of around \$100 million. (Google, 2006)

One of the enterprise division's latest products is Google Apps, an online application suite that targets mainly small businesses, but also larger companies and academic institutions (Google, 2007a). Google Apps consists of an e-mail and instant messaging application, an online calendar, a personalized homepage, a web site creator and a simple word processing and spreadsheet application. These applications are interconnected to different degrees. For instance, an attachment received in Gmail, the e-mail application, can directly be opened in Google Docs, the word processor. Google's locally installed desktop components – its browser toolbar and its PC-based search engine – also contain some links to Google Apps.



**Figure 19: Google's e-mail application with advertisements**

For individuals, Google Apps are free. However, Gmail uses advertising to generate some revenue in this free edition. Businesses can customize some aspects of Google Apps and get more online storage space for an annual fee of \$50 per user.

Google doesn't disclose the number and type of customers that use Google Apps. However, the company has indicated that some large companies, including General Electric and Procter & Gamble, are evaluating the suite.

Google seems to intend a further extension of its application suite. It acquired wiki vendor JotSpot in November 2006 and is expected to integrate this product into Google Apps (Delaney, 2006).

### 5.4.10 SAP

For a profile of SAP, please see section 3.2.3.4.

SAP, the market leader in the ERP market (AMR Research, 2006), was relatively slow to respond to the increasing demand for SaaS applications. In February 2006, it introduced a first partly hosted product with its "SAP CRM on-demand" solution, which is an adapted version of its traditional "mySAP CRM" package (SAP, 2006). SAP didn't disclose revenue figures for this product in 2006.

In March 2007, SAP started to demonstrate to selected customers a new SaaS suite code-named "A1S" (Blau, 2007). A1S will be a hosted application suite for mid-market cus-

tomers offering ERP, CRM and supply chain management functionality. As a main selling point, A1S is designed to integrate with SAP's flagship product "mySAP ERP". The main goal of this integration is apparently to sell a simpler, cheaper hosted product to subsidiaries of large companies already using SAP's software. SAP CEO Henning Kagermann acknowledged the fact that the introduction of a subscription model was a major step for the company, particularly since upfront costs were significant (Wiesmann, 2007).

#### **5.4.11 Oracle**

For a profile of Oracle, please refer to section 3.2.3.3.

Oracle has been offering an "On Demand" product line for several years (Chou, 2005). Under this brand, it sells managed application services. The software can be either hosted remotely by Oracle (or a partner company), or it can run on customer premises, while Oracle manages the system remotely.

The On Demand product line includes hosted versions of Oracle's E-Business and Collaboration suites, its PeopleSoft and JD Edwards ERP products, and its Siebel CRM product. Of these, only Siebel CRM is currently available with a pure subscription-based model.

According to its annual report, Oracle generated revenues of \$398m with its on demand offerings in 2006, about 2.6% of its total revenue (Oracle, 2007). This percentage remained approximately constant from 2004 and 2005. The revenue figure included subscription-based products, managed services and additional infrastructure-related services.

Apparently, hosted applications are not a particularly large or rapidly growing part of Oracle's core business at this point, although CEO Larry Ellison suggested in an interview that they are highly profitable (Waters, 2006).

#### **5.4.12 Summary of Case Studies**

The table on the following page summarizes the companies described above. The following criteria are used:

- Year founded
- 2006 revenue, if available
- Is the SaaS product line the main source of revenue, i.e. is the company focused on web-based products?
- What types of products are offered?
- What is the product strategy?
  - Niche = concentrates on one very specific application
  - Focus = concentrates on one broader application, such as CRM
  - Suite = offers a whole range of applications

- Is the product usable in an offline setting, i.e. when the user does not have Internet connectivity?
- Does the product offer integration with common desktop applications, such as Microsoft Office?
- Does the product offer programming interfaces that enable it to be connected to other applications?
- Is the pricing transparent?
  - Fully transparent = Full price list is available online
  - Partly transparent = Company discloses minimum pricing, but final price depends on customer situation
  - Not transparent = Customer-specific pricing, only disclosed when company is contacted.

	Salesforce.com	RightNow	Netsuite	Xactly	Smartsheet.com	37signals	Zoho/AdventNet	Google	Microsoft	SAP	Oracle
Year founded	1999	1997	1998	2005	2005	1999	1996	1998	1975	1972	1977
2006 revenues	\$497.1m	\$110.39m	N/A	N/A	N/A	N/A	N/A	\$10.6b	\$44.3b	\$9.4b	\$14.4b
SaaS product is main source of revenue	Yes	Yes	Yes	Yes	Yes	Yes	no	no	no	no	no
Types of SaaS application	CRM, sales force management	CRM, sales force management	ERP, CRM	Sales force compensation	Project management	Project management, CRM	e-Mail, word processing, spreadsheet, presentation, application development, CRM, project management, conferencing	e-Mail, word processing, spreadsheet, web page design	CRM, project management, web page design	CRM; ERP under development	CRM; ERP with limitations
Product strategy	Focus	Focus	Suite	Niche	Niche	Suite	Suite	Suite	Focus	Focus	Focus
Offers offline functionality	Yes	Yes	Partly	No	No	No	No	No	Partly	No	Yes
Integration with desktop applications	Yes	Yes	Yes	No	No	Limited	Yes	Limited	Limited	Yes	Yes
Offers programming interface	Yes, full platform	Yes, limited	Yes	Yes	No	Yes, limited	No	Yes, limited	No	Yes	Yes
Pricing transparency	Partly transparent	Not transparent	Not transparent	Not transparent	Fully transparent	Fully transparent	Fully transparent	Fully transparent	Fully transparent	Not transparent	Not transparent

Table 14: Summary of case studies

There are a few observations that can be made based on these case studies:

- CRM clearly seems to be the application type of choice for most SaaS-oriented companies. It is not only the most popular application type in the overall sample (see 5.3.1.1), it is also the field where the currently most successful companies are active. Even the established traditional software companies (Microsoft, SAP, Oracle) start their SaaS offerings with CRM.
- There is no clearly winning product strategy, but a focus on a specific broader application (such as CRM) seems currently to be the most promising strategy. Both strongly concentrated niche strategies and broader suites are still unproven at this point.
- The more advanced SaaS companies offer at least part of their functionalities also as an offline component for times when users do not have an Internet connection. This is partly explainable due to the fact that salespeople (the main target users of most CRM offerings) tend to be on the road and away from broadband connections at least some of the time. Only the younger companies require a broadband connection at all times for their products' users.
- Most companies offer at least some level of integration with desktop applications. The same is true for programming interfaces that enable the integration with other software. Most advanced companies offer extensive APIs (Application Programming Interfaces), some offer limited interfaces using simple XML-based web services. Quite obviously, SaaS applications don't exist as standalone applications, but have to be integrated with other programs.
- The trend towards more transparent pricing is quite interesting. Traditionally, enterprise software pricing is pretty intransparent. A prospective customer has to discuss his or her specific situation and needs with a sales representative before receiving a quote. This is profit-maximizing behavior on the part of the vendor: Since software is an information good with marginal costs of almost zero, a maximum of price discrimination optimizes a software company's profits (Shapiro and Varian, 1998). It is therefore best for a software company to find out as much as possible about a customer's willingness to pay. On this basis, the customer is charged an individual, profit-maximizing price whenever possible.

This approach is still used by the more traditional companies, such as SAP or even RightNow. They don't disclose any pricing information at all. In contrast, many of the younger companies at least publish minimum pricing or even a fully transparent price list.

This apparent increase in transparency could have two reasons: Firstly, many customers nowadays know that software prices are typically subject to negotiation. By establishing fixed and transparent prices, a vendor can reduce the pressure to grant discounts. Secondly, many web-based applications are sold without any human intervention on the part of the vendor. For instance, customers can sign up for 37signal's applications or Microsoft Office Live without ever talking to a sales



representative. This is only possible when prices are communicated openly. This high degree of automation saves costs, but prevents software vendors from achieving optimal price discrimination. To some extent, vendors are trying to achieve price discrimination by offering a larger number of product versions at different price points.

This limited selection of companies shows clearly that there are still many different strategies in the SaaS segment, but also that some common patterns are emerging.

The financial information available from the few public SaaS companies seems to suggest that this model can generate attractive gross margins, but that marketing and sales costs still consume most of the profits due to the early stage of the market.

A final important observation is the fact that practically all established software companies are at least experimenting with web-based software. Of the leading companies listed in section 3.2.3, only Autodesk currently doesn't offer any major fully web-based products, which is explainable by the data-intensive nature of its applications. Quite apparently, web-based software is approaching the mainstream of the vendor market.

## **5.5 Customer Adoption of Web-Based Enterprise Software**

Most observers agree that Web-based software is a significant development for the software industry overall and will have a major impact on the market for software products. However, there is still a lot of disagreement about how far web-based software has been adapted yet, and differences in the predictions for future market development are even greater.

### **5.5.1 Current Adoption and Market Size**

There are many studies and surveys about the adoption of web-based software (or SaaS, or on-demand software), but it is difficult to get a consistent picture of the market situation, since definitions, methodologies and the quality of samples vary significantly.

However, some study results give an impression of the current state of the market:

#### **Customer Adoption and Awareness:**

- A 2004 study by Summit Strategies (CRN, 2004) found that 31% of enterprises were already using SaaS, and another 28% were evaluating or planning to evaluate SaaS-based solutions. For small companies, the percentages were even higher, with 36% using SaaS already and 31% planning to do so.
- A January 2006 study by Saugatuck Technology (Smith, 2007a) indicated that 28.4% of large enterprises plan to use SaaS for applications with a widely distributed workforce (i.e. collaborative applications), but only 12.5% planned to use SaaS for mission-critical applications.  
One year later, the numbers in a similarly structured survey had risen dramatically: Now, a full 70% of large enterprises planned to use SaaS for collaborative applications, and 53% considered it even for mission-critical applications.

- Another 2007 study by Saugatuck (Smith, 2007b) stated that 26% of survey participants at the beginning of 2007 had one or more SaaS solutions in place, versus only 11% a year earlier. Only 8% of respondents said they were not planning to deploy SaaS.
- Gartner, Inc. said in a 2006 report that in 2006 “90 percent of SaaS deployments are for departmental use, providing baseline capabilities that deliver little, if any, competitive differentiation, nor are they designed to do so. The functionality is achieved with minimal (and sometimes no) intercession by IT.” (Maoz, 2006).

### **Market Size:**

- Gartner, Inc. estimates that “in 2005, approximately 5% of business software was delivered as SaaS.” (Desisto et al., 2006).
- The Software 2006 Industry report by McKinsey and the SandHill Group estimated the SaaS market to be around \$6 billion in 2006, based on IDC figures (Berryman et al., 2006). This would be still a tiny fraction of the global market for software products and services, which is estimated to be well above \$1 trillion (Cusumano, 2004).
- According to Jupiter Research (McGeary et al., 2005), hosted CRM applications had a market share of 19% in 2005, while packaged software accounted for 33% and “home grown applications” for 47%.
- Forrester Research estimated the total global market for CRM software to be worth \$12 billion in 2005 (Band, 2006), which is consistent with other estimates by AMR Research (Waxer, 2006). This would indicate that the two leading SaaS CRM vendors, Salesforce.com and RightNow Technologies, achieved a combined market share of 3.3% (based on 2005 revenues). However, a Gartner Dataquest report estimated Salesforce.com’s market share alone to be 4.9% in 2005 (Gartner Dataquest, 2006)

To summarize: The most reliable and tangible market figures indicate that vendors of web-based CRM solutions had a market share slightly above 5% in 2005. Since CRM is one of the strongest fields in web-based applications, the overall market share of web-based solutions of the total enterprise software market is probably significantly below 5%. When measured against the total size of the global software market, SaaS is still a very small sector, accounting for less than 1% of the total market.

However, mindshare with users seems to be significantly greater than that. All surveys consistently indicate a large and rapidly growing willingness to adopt SaaS or at least evaluate this type of application. Most studies suggest an adoption rate of between 10-30% of respondents.

A possible explanation for the apparent discrepancy between users’ cited adoption rate and the share of total revenue is the fact that hosted solutions are often significantly cheaper than installed software. Also, since new users do not generate a revenue spike for SaaS companies, but rather increase the revenue base over time, a fast growth rate in user

numbers is less noticeable financially than in the case of license-based software vendors. Furthermore, the professional service revenues generated by SaaS are usually significantly lower.

### **5.5.2 Predictions**

Many studies predict significant growth rates for hosted applications. To cite just a few:

- A survey by McKinsey concluded that 61% of large U.S. companies plan to implement SaaS applications over the next few years (Carr, 2006).
- Saugatuck Technologies predicts that by 2010, 21% of users will adopt SaaS for business applications, 24% for infrastructure applications, and 55% for other applications (Bednarz, 2006).
- Gartner estimates that “in 2010, 30 percent of software revenue will be derived from software delivered via SAAS models.” (Pring, 2005).
- In another estimate, Gartner predicts the SaaS market to reach \$19.3 billion by 2010 (Sims, 2007).
- The Software 2006 Industry report (Berryman et al., 2006), already cited above, expects a 20% annual growth rate for SaaS, growing to \$10.9 billion in 2009.

Quite obviously, these estimates differ dramatically, particularly with respect to the estimated market size. This is probably not untypical for a still very young segment of the IT industry.

## **5.6 Disruptive Characteristics**

One of the most frequently cited strategic frameworks for the analysis of innovations is Clayton Christensen’s concept of “disruptive technologies” (Christensen, 1997). Disruptive technologies, according to Christensen, often cause the decline of incumbent companies and the rise of new competitors. Some of the classic examples are the replacement of mainframe computers with PCs, tabletop photocopiers, or Dell’s direct sales model (Christensen, 2003).

Christensen describes three defining characteristics of markets that are facing a disruptive innovation:

1. A “rate of improvement that customers can utilize and absorb” (Christensen, 2003).
2. A “distinctively different trajectory of improvement that innovating companies provide as they introduce new and improved products” (Christensen, 2003). Typically, these incumbent companies overshoot their customers’ requirements after a certain point.
3. A “distinction between sustaining and disruptive innovations” (Christensen, 2003). While sustaining innovations target existing customers who have very demanding requirements, disruptive innovations introduce less capable, but much cheaper products into a market. Disruptive technologies therefore are not attractive to high-end users, but rather target low-end segments or entirely new markets. For this reason, the incumbent high-end producers initially do not have an incen-

tive to compete directly against the new entrants and, in the long term, end up losing more and more market share.

It is obviously important for incumbent companies as well as for new entrants to think about the disruptive potential of a new technology or business model. When Christensen's criteria are applied to the enterprise software market and the potentially disruptive web-based applications, the following observations can be made:

There is clear evidence about the increasing discrepancy between customer requirements and the technology that incumbent vendors are trying to sell. Analysts report a growing "software value gap" (Snyder, 2006) in customers' perception of the business value generated by increasingly complex and expensive software. Many customers feel that their existing software systems are adequate and see little reason to upgrade their systems to the latest versions offered by software vendors – a "good enough" crisis: A situation in which product-based differentiation is no longer rewarded, thus triggering the maturation of every product category" (Vincent et al., 2005).

The wave of new entrants offering web-based solutions therefore could be a response to this crisis of the traditional enterprise software market. However, in order to have a truly disruptive effect, web-based applications would need to show two important characteristics.

Firstly, they would need to be limited to clearly inferior performance and therefore not be attractive to incumbents' best customers. This is probably true to some extent. As explained in section 4.3, there are still numerous drawbacks that users of web-based applications have to accept. The most demanding customers of enterprise software probably will not opt for a web-based solution at the current state of the technology.

Secondly, the new technology would need to be significantly cheaper than existing solutions. It is not really clear if this is true for web-based applications.

A simple comparison between the fees for a SaaS solution and the often hefty license price for a traditional enterprise application does not reflect overall costs adequately. Some SaaS vendors charge additional set-up costs or additional fees for storage and advanced features. Furthermore, SaaS fees are typically only locked in for the duration of the contract and later could be raised by the software company (Bona, 2004). It is therefore difficult to reliably estimate the long-term costs of a web-based solution and even more difficult to find truly significant cost advantages compared to traditional software.

Furthermore, the total payments to the software vendor that users of SaaS solutions have to face are not necessarily smaller than in the case of on-premise software. The following table illustrates the total software costs that a customer of Salesforce.com faces in comparison to a traditional CRM solution by Siebel Systems (now part of Oracle; the comparison is based on data from 2004, the last year of Siebel's independence).

	<i>Salesforce.com (2006)</i>	<i>Siebel (2004)</i>
Users at end of year	646,000	2.9 million
Users added this year	247,000	700,000
New license revenues	-	\$ 487 million
Maintenance and subscription revenues	\$451 million	\$469 million
Average license revenues per new user	-	\$696
Average annual subscription and maintenances revenues per existing user	\$699	\$213
<b>Total software costs per user for 3 year period</b>	<b>\$2,097</b>	<b>\$1,335</b>

**Table 15: Comparison of software costs for Salesforce.com and Siebel CRM**

Hence, for a three year period, the license and maintenance costs for an average user would add up to \$1335 in the case of Siebel, significantly lower than Salesforce.com's subscription fee for the same time range. These calculations are only a rough approximation and of course do not include costs for infrastructure and integration work (which are generally assumed to be significantly higher for Siebel users, since the software is installed on customers' own server infrastructure).

Furthermore, cost savings for IT infrastructure and maintenance are the most frequently cited advantages of SaaS. However, SaaS requires a stable and fast broadband connection that can be fairly expensive, and savings on desktop PC infrastructure are probably negligible.

The true total cost of ownership (TCO) of a web-based solution therefore has to be calculated individually for every customer situation, which can be quite complex (Rosenberg and Wright, 2007). While SaaS certainly has some obvious advantages, such as a faster rollout time, a lower TCO is not automatically a given.

It is furthermore important to monitor the reaction of incumbent vendors to the new technology. Almost all traditional vendors are currently trying to come up with web-based products as a response to the new entrants. However, this fact in itself does not negate the disruptive nature of web-based applications. Historically, most incumbents have tried to react in such a way when facing a potentially disruptive innovation (Utterback, 2004). A well-known example is the IBM PC that enabled IBM to participate in the PC market, only to finally exit this business in 2004 after years of losses (Bulkeley, 2004). The question is if the incumbents have enough motivation to really adapt to the new wave of technology (Christensen, 2003). Given the highly profitable traditional business of the leading enterprise software vendors, this is probably doubtful. In addition to this, web-based

software requires new skill sets (as explained in section 4.3) that are not necessarily easy to build for incumbent companies.

Due to the factors mentioned above, it is more likely that SaaS will turn out to be a new-market disruption, as opposed to a low-end disruption. Christensen describes new-market disruptions as having a “[lower] performance in ‘traditional’ attributes, but *improved performance in new attributes – typically simplicity and convenience*” (Christensen, 2003). This description fits web-based applications quite well. The advantages of SaaS are particularly attractive to customers that do not yet use a particular application type (e.g. those that do not yet have a CRM system). By contrast, overserved users of traditional enterprise software still have the option to stay with older versions of their already installed applications.

To summarize: web-based enterprise applications show most characteristics of a disruptive innovation, but it is unclear if their cost advantage is significant enough to trigger a strong migration to this new technology. It is more likely that web-based enterprise software will be attractive to non-users of a specific application type than to existing users seeking a lower-cost option. Therefore, web-based solutions are most likely a new-market disruption in Christensen’s terminology.

## 5.7 Emerging Dominant Designs

Another crucial concept for the analysis of innovative markets are “dominant designs” (Utterback, 1994). In the early stages of a technology, there are many experimental product variations, business models and processes. Many new market entrants try to find a winning combination, since barriers to entry are still low.

“Within this rich mixture of experimentation and competition some center of gravity eventually forms in the shape of a *dominant product design*. Once the dominant design emerges, the basis of competition changes radically, and firms are put to tests that very few will pass. Before long, the ecology of competing firms changes from one characterized by many firms and many unique designs, to one of few firms with similar product designs.” (Utterback, 1994)

The market for traditional enterprise software clearly found a dominant design years ago. Most companies sell client/server-based software (with PCs as client machines), based on a perpetual license. In addition, customers typically have to pay a maintenance fee of 15-20% of the original license price. The current consolidation of this market (see section 3.2.4) is a classic example of the phenomena described by Utterback.

In contrast, the large number of new entrants into the market for web-based enterprise software (see section 5.3.1.4) seems to suggest that this new market is still in its infancy. Furthermore, the large number of business model variations (see 5.3.1.2) points to the absence of a dominant design in this phase of market development. However, a purely subscription-based model, in some cases combined with professional service offerings, seems to be slowly emerging as the dominant design for the revenue model.

In terms of technical designs, there is still no clearly dominant model. For instance, the 11 companies examined in the case studies of section 5.4 use six different server-side programming environments, with only Java Server Pages used by more than two compa-

nies. Furthermore, in terms of approaches to application hosting (on vendor's own premises vs. outsourced) there still are several different patterns.

The variation in client-side techniques is not quite as big, with AJAX<sup>4</sup>-based interfaces now used by most vendors. AJAX is probably what Utterback called a “satisficer” (Utterback, 1994): Not the technology offering the best technical performance, but the one good enough for the largest group of users (Maoz, 2006).

To summarize, there is no dominant design yet for web-based enterprise software. The market still offers many variations in pricing models, client- and server-side technologies, deployment options, programming interfaces, and so on. The only design elements slowly emerging as potentially dominant are subscription-based pricing (with a monthly fee) and AJAX-based user interfaces.

---

<sup>4</sup> Asynchronous JavaScript and XML

## **6 Strategic Framework**

In order to develop a forward-looking perspective on web-based enterprise software, this chapter will try to establish a strategic framework from both the vendor perspective and the customer perspective.

It's obvious that software vendors have to think strategically about web-based software, whether they are incumbent traditional software companies or startups. The emerging market for web-based software is already impacting the software industry in a significant way, as discussed in the previous chapter, and it is likely that this impact will only grow over time. A long-term perspective that integrates this new type of application into an overall strategic plan is therefore crucial for success.

On the other side, potential and existing customers of web-based enterprise software have to think in an equally strategic way about this new type of application. There are many fundamental considerations about what types of applications should be outsourced to a SaaS vendor: data confidentiality, risk mitigation, the role of the internal IT department, and so on. On an operational level, every single user in a customer organization is directly affected by a shift to web-based applications. Therefore, this wave of innovation is significant on many levels and needs to be addressed strategically beyond the IT department.

### **6.1 The Vendor Perspective**

The strategic framework for vendors presented here will focus on specific aspects that are relevant to a SaaS strategy. It will not, however, touch on very high-level strategic topics.

As explained by Michael Cusumano in his book "The Business of Software" (Cusumano, 2004), generic strategic concepts such as Porter's "Five Forces" (Porter, 1980), Hax and Wilde's "Delta Model" (Hax and Wilde, 2001) or Christensen's concept of "disruptive innovations" (Christensen, 1997) are very useful for every type of company, including software companies. However, the more specific and immediate strategic challenges for a software firm affected by the move to web-based applications need to be analyzed on a more detailed level.

The analysis is divided into three dimensions:

- Customers and Markets
- Business Model
- Technology

#### **6.1.1 Customer and Market Strategy**

##### **6.1.1.1 Target Customers**

A first crucial decision is what customer groups to target. There are several relevant criteria for the identification of target customer segments:



- Should the product be sold to companies or consumers?  
(Due to the focus of this thesis, only corporate customers will be explored further)
- Customer company size (small, medium-sized, or large businesses)
- Customer industries
- Degree of specialization of customer (functional, industry), i.e. is the target market a niche or a mass market?

The choice of customer segment is highly interdependent with other strategic decisions. For instance, a product that targets large enterprises has to offer much more advanced integration capabilities than a product that is sold to very small companies.

#### **6.1.1.2 Geographic Strategy**

At first sight, since the Internet is a global medium, the market for a web-based application is automatically global. However, for many types of applications and customers, this is not automatically true, and international success requires considerable effort on the part of the software company.

First of all, customers expect their software to be offered in their local language. Most popular desktop software (e.g. Microsoft's products) is available in dozens or even hundreds of languages, and customers have come to expect that they can buy localized versions of software. Therefore, vendors of web-based applications will have to offer different language versions if they want to be successful internationally.

Secondly, for some types of applications, local regulations are highly relevant. A typical example are VAT rules that are slightly different in every country. A CRM or accounting program that is not able to handle these differences is not very useful for international customers.

Thirdly, in the author's experience, there are often many small and seemingly trivial details that prevent software from being used in other countries. A classic example is the fact that postal codes do not have five digits in every country in the world, but often American software has hard-coded routines that test for five digits in order to determine if a ZIP code is correct. This kind of detail problem is annoying to both customers and the vendor, and therefore should be taken into consideration from the start of development.

For these reasons, a software vendor has to plan from the beginning which countries he wants to sell his software to. After the first development phase, promoting and selling the software in other countries is not trivial and often requires the establishment of local offices. For instance, Salesforce.com currently has sales offices in the US, Canada, Mexico, Japan, Australia, Singapore, Great Britain, France, Germany, Switzerland, Sweden, Italy, Ireland, the Netherlands, Belgium, and Spain (Salesforce.com, 2007). It is obviously not true that web-based software can be sold exclusively over the Internet.

#### **6.1.1.3 Channel Strategy**

In the traditional software model, target customer segments correlate quite closely with sales channels. For instance, enterprise software for large companies is typically sold directly to customers, maybe with the help of a consulting firm, while software for small businesses is typically sold through retail channels or value-added resellers.

For web-based software however, this clear distinction between channels does no longer apply. Any type of customer can potentially sign up for a web-based application, since the delivery mechanism is basically the same for every customer. There is also no need for a channel partner that sets up server infrastructure on customer premises.

The role of the sales channel is therefore diminished to some extent. However, many of the bigger SaaS companies still work with channel partners that carry out some of the sales work and particularly help customers to set up and customize the software product.

For example, Salesforce.com has a list of “consulting partners” that includes major systems integrators such as Accenture and Cognizant. Furthermore, it has a “referral partner” program that offers commissions to companies that refer new customers to the CRM vendor (Salesforce.com, 2007). RightNow Technologies uses a similar structure, but in addition has OEM partners that resell RightNow’s software in connection with their own products. 37signals has an “affiliate program” that pays existing users for the referral of new customers. Depending on the level of subscription, referrers can earn up to \$50 per new customer.

Vendors therefore need to decide if and how to integrate a sales channel, and what the commission structure should be.

#### **6.1.1.4 Service and Support Strategy**

Closely connected to the channel strategy is the question how service and support should be provided to customers.

There are two basic strategies:

- The vendor is the primary source for customer support via phone hotlines, online help systems, and field service staff.
- Support is provided primarily by third-party service providers, such as IT systems integrators.

Obviously, providing service in-house can generate additional revenue, but is likely to increase costs. Working with external service partners can be positive for the software vendor’s gross margin, but reduces control of customer accounts and can cause quality problems if service partners don’t operate as desired.

In addition to these support services, many vendors use additional support channels, such as online discussion forums, knowledge bases, and user groups.

#### **6.1.1.5 Strategy for “Crossing the Chasm”**

As explained in section 5.5, customer adoption of SaaS-based applications is still in its early stages. In the terminology of Geoffrey Moore, web-based software overall is probably still in the early adopter phase. As Moore describes in his classic book “Crossing the Chasm” (Moore, 2002), the toughest challenge for technology firms is often to cross the chasm between early adopters and the early majority of customers. While the early majority is motivated by the strategic vision about what a new technology can do, the early majority demands tangible business benefits.

Crossing the chasm is probably not equally difficult for every type of web-based application. For instance, most customers should by now be familiar with the concept of CRM and probably recognize that CRM systems could be beneficial for their business (Waxer, 2006). Web-based CRM can offer tangible cost savings over traditional solutions and therefore addresses the main concerns of an early majority. It is therefore not particularly surprising that the currently most successful SaaS firms sell CRM software.

The situation is different, however, for other application types. Collaborative software, such as wikis or web-based project management, is still a fairly new type of solution, and therefore the vendors of these products will have to work hard to cross the chasm.

Therefore, it is to some extent probably helpful for SaaS startups that established software vendors are starting to promote these new types of applications themselves, although their products often can't match the quality of the best solutions in the market. According to a recent survey by Forrester Research, most CIOs would be more interested in collaborative software "if it would be offered by a major incumbent vendor such as Microsoft or IBM" (cited by MacManus, 2007). An increase in overall acceptance of collaborative applications could open opportunities for smaller, best-of-breed vendors.

## **6.1.2 Business Model Strategy**

### **6.1.2.1 Application Type**

A crucial decision for every vendor is which types of applications to offer to target customers. As discussed in section 5.4.12, there are three basic strategies:

- Concentrate on a single niche application.
- Focus on a single, but broader (horizontal) application type, e.g. CRM, ERP, collaboration.
- Offer a suite of different application types.

This choice of basic strategy is, apart from the choice of a particular customer segment, probably the most crucial decision for a software vendor. It influences all other subsequent decisions and impacts the economics and risk structure of a company significantly.

As a secondary decision, a vendor has to decide which specific application types to offer. This decision is obviously interdependent with several other strategic dimensions.

### **6.1.2.2 Deployment Options**

Since this thesis focuses on web-based applications, we assume that solution deployment over the Internet as a hosted application is a given for the type of vendor examined here. However, in many cases a combination with other deployment options can make sense. As analyzed in section 5.3.1.3, more than a quarter of vendors combines hosted solutions with other deployment models, such as locally installed software or hardware appliances.

The most important driver for this decision are clearly customer requirements, particularly concerns about data confidentiality and system availability. Some customers (according to some studies, the majority – see section 5.5) demand mission-critical software solutions to be installed on their premises. Several vendors of web-based products accommodate this wish with alternative, on-premise versions of their software.

Generally speaking, offering multiple deployment models is likely to increase a vendor's costs significantly, but it might be the only way to win customers that don't fully feel comfortable with hosted applications.

In some cases, it can be necessary to offer just certain components of the overall solution in a different deployment model. For instance, several sellers of web-based CRM solutions offer locally installed client software that enables traveling salespeople to use the software even when they are not connected to the Internet.

Vendors need to carefully consider the trade-offs between customer acceptance and the different economics of multiple deployment models.

### **6.1.2.3 Revenue Models and Pricing Strategy**

As in the case of deployment models, many vendors use several variations of revenue models and pricing strategies. Sometimes, a revenue model is connected directly to the mode of deployment. For instance, some companies sell software as a hosted version based on a monthly fee, but also as a locally installed software product for an upfront license price.

But even just for a hosted software product, there are many different pricing models that a vendor has to think about. Two perspectives have to be considered: What kind of pricing is acceptable to the customer, and what is the ideal pricing for the vendor?

Some of the important variables for a SaaS-based pricing are:

- Should the subscription fee be billed monthly or annually?
- How long should the contract duration be?
- Which different versions of the product should be offered, and at which price points? What are the best product variables that allow a pricing differentiation?
- Should there be a basic free version of the product that helps to attract customers?
- Should the pricing be fully transparent (listed on the website and in marketing material), partly transparent or only based on customer-specific negotiations?
- Should there be a discount scheme for larger customers, and what should be its structure?

In addition to SaaS revenues, a vendor of web-based applications has other options to generate revenues:

- Premium support services
- Professional services for the set-up, customizing and integration of the solution
- Training services
- If a development or integration platform is offered: Royalty payments from third-party developers

There are however pricing components that are customary in the traditional enterprise software world, but not generally accepted in a SaaS model. Examples include: Additional software maintenance fees (typically 15-20% for traditional software) and additional license fees for major product upgrades. Pricing for SaaS is also almost always based on named users, i.e. individual user accounts. Other pricing schemes that can be found in traditional software, such as pricing for concurrent users, or flat fees for whole enterprises, are far less common in SaaS.

#### **6.1.2.4 Target Revenue Mix**

One of the most important strategic variables for a traditional software company is the mix between product and service revenues (Cusumano, 2004). In a SaaS-based business model, this distinction is typically made between subscription revenues (i.e. the monthly or annual fees that customers pay) and professional services revenues (consulting fees paid for support, training, installation, integration etc.).

For traditional on-premise software, gross margins on product revenues tend to be very attractive, often well above 90%. The reason is that the manufacturing and distribution of the media carrying the software product is very cheap (Shapiro and Varian, 1998). Therefore, software companies with a higher share of product revenue tend to be more profitable overall than service-heavy companies.

The same is true for SaaS-based business models, although not quite to the same extent. As shown in section 5.4, gross margins for subscription revenues tend to be around 85% at SaaS-only companies such as Salesforce.com or RightNow Technologies. The reason is that the costs of hosting infrastructure and hosting operations reduce the gross margin. Furthermore, a basic level of tech support is typically included in subscription fees, and the associated costs reduce the margin further. Still, it seems to be more attractive to have a high share of subscription revenue vs. professional service revenue. It is therefore understandable that SaaS companies try to build a network of service partners.

It remains to be seen if SaaS companies will experience the same effect as their traditional counterparts once their business matures: Many established software companies see an increase in the share of service revenues as they grow and their market gets saturated (Cusumano, 2004). A logical explanation for traditional software companies is the fact that they can sell less new licenses in a maturing market. Furthermore, prices for licenses often fall over time.

The situation is different for SaaS-based models: new customers don't generate an initial spike in revenues, but rather increase the overall long-term revenue base for subscription revenue. At the same time, revenues from professional services probably rather decrease over time: Once the system is customized for a customer, there's only little need for further consulting work, and there is no necessity to install new versions of the software on customer infrastructure. The SaaS model therefore could prove to be more attractive in the long run than the traditional model of installed software.

However, since SaaS is still a very new model, there is no experience base yet about the costs associated with operating SaaS solutions over long periods of time. It is not clear if SaaS companies can keep up their service cost advantage over time, or if they will be affected by the same adverse effects that traditional software companies experience when

moving from one generation of a product to the next. SaaS products are getting more and more customizable, and typically, migrating customized functionality to a new major version of the underlying product can be costly for both customers and vendors.

Furthermore, the time lag between product generations can be significant. For instance, SAP's R/3 ERP product was released in 1992, more than a decade after its predecessor R/2 (SAP, 2007). The company today is still offering migration services that help customers to move from R/2 to R/3. In comparison, the most experienced specialized SaaS companies were only founded in the late 1990s and therefore are typically still selling their first major product generation.

It therefore remains to be seen if the SaaS model turns out to be more robust against such migration effects than the traditional on-premise software model. Although the SaaS model clearly is more efficient for minor version upgrades, there is no obvious structural advantage for the migration between major product generations.

#### **6.1.2.5 Platform Strategy**

Platforms and the “ecosystems” around them are a major element of the software industry. Some of the largest and most profitable IT companies built their success on platform leadership (Gawer and Cusumano, 2002).

Web-based software is often associated with the notion of the Internet as “the ultimate platform” (Vincent et al., 2005). Proponents of this view argue that the Internet, through its ubiquity and open standards, actually has become the IT platform of choice, open to everybody and owned by nobody. Previously important platforms, such as desktop operating systems, are becoming increasingly irrelevant in the Internet age.

The standards setting process for the Internet is organized by non-profit organizations and is deeply democratic in its nature (IETF, 2007). However, almost all Internet standards focus on technical aspects only and are typically fairly low-level in nature. For instance, the standards proposed by the World Wide Web Consortium (W3C, 2007) concentrate on elements such as the HTML web page markup language, meta-standards such as XML, and generic guidelines for user interfaces. There are no application-specific standards that specifically target business applications and business logic (Vincent et al., 2005). For that reason, there is still some room left for the establishment of application-oriented platform leadership in the era of web-based software.

There are already some companies that are trying to do this. Salesforce.com, through its AppExchange platform and its Apex programming language (see 5.4.1), is clearly trying to set a standard for CRM-like business applications. There are also smaller companies, such as Seattle-based Etelos (Etelos, 2007) that are trying to achieve the same goal. Google, Microsoft, Yahoo, Amazon.com and eBay all offer open programming interfaces to some of their online services. Currently, these interfaces are mostly focused on consumer-oriented applications or very specific purposes (such as the management of auctions on eBay), but the potential for further extension is obvious. For instance, Amazon.com through its Web Services division is already offering advanced infrastructure and computing services to third-party developers. Of course, all these services easily interact with its popular online store (Amazon Web Services, 2007).

Vendors of web-based software therefore have to make a similar decision as their traditional counterparts (Gawer and Cusumano, 2002): Does the company want to be a platform leader or a complementor? If it tries to be a leader, what is the scope of the platform, and how can it attract complementors?

For complementors, the crucial decision is which platforms to support or build on. Since the market is still quite immature and developing rapidly, the risk of this decision is considerable.

### **6.1.3 Technology Strategy**

#### **6.1.3.1 Development Platform**

One of the most basic decisions a traditional software vendor has to make is what technical platform to develop for. For instance, vendors of ERP software for small businesses have to decide whether to develop just a version for Microsoft Windows or also for Apple's Macintosh, Linux and other platforms. Furthermore, the company has to decide which database to use, which programming language to code in, and which middleware products to build on. All of these decisions impact how many customers can run the program on their infrastructure and hence the potential market for the product.

This type of decision doesn't have the same importance for vendors of web-based software. Customers of a web-based application are not impacted by the type of server platform used by the vendor, since they interact with the system over standardized Internet protocols. The vendor therefore is free to implement his software on whatever platform he thinks is the most efficient.

However, there are some minor considerations for the implementation of the client side of a web-based application:

- What browser types and versions should be supported? Not all older browser versions are able to support modern user interface concepts. Supporting many browsers can increase costs for development and testing.
- What type of client-side runtime environments (if any) should be used? For instance, some applications require client-side Java or Adobe Flash. This can increase functionality, but lead to support and compatibility issues.
- Are there any locally installed client modules, e.g. toolbars, that could enhance the application's functionality?

#### **6.1.3.2 Hosting Strategy**

Since web-based applications are remotely hosted, the performance of the necessary hosting infrastructure is of crucial importance for this business model. At the same time, running such an infrastructure is a major cost factor.

There are four basic strategies that a software firm can pursue for its application hosting:

- Build and fully manage its own infrastructure.
- Physically locate the infrastructure at an outsourcing data center, but manage servers remotely.

- Fully outsource hosting operations (including systems management) to another provider, but sell it under own label.
- Fully outsource hosting to third parties (equivalent to the ASP model described in 4.1)

Several factors influence this decision:

- Performance requirements, both in terms of server performance and network bandwidth
- Requirements about availability, security, and redundancy in case of disaster
- Legal requirements and audit standards (such as SAS 70 Type II certification)
- Desired cost structure and risk sharing

### **6.1.3.3 Integration Strategy**

Particularly larger customers of web-based software often have the need to integrate a hosted solution with other software systems, both externally hosted and installed on customer premises. A software vendor needs a strategy on how to accommodate these requirements.

Some essential elements of an integration strategy are:

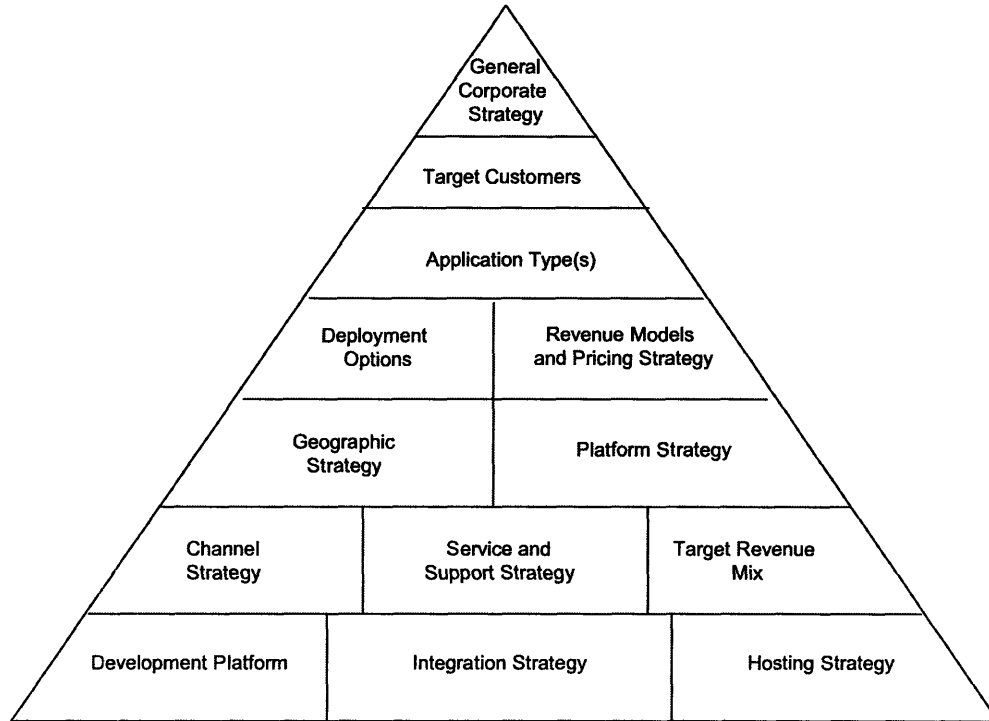
- The type of general architecture used. A service oriented architecture (SOA) is currently the most typical approach.
- Types of middleware (if any).
- If and how to bill customers extra for integration mechanisms.

### **6.1.4 Summary**

The strategy for a vendor of web-based enterprise software consists of several fairly complex choices that are strongly interdependent. It is therefore not surprising that the vendors in today's market are still experimenting with very different approaches, as documented in chapter 5.

As a graphical representation, the following pyramid illustrates the implicit hierarchy of strategic decisions for a vendor of web-based enterprise software:





**Figure 20: Strategy pyramid for web-based enterprise software<sup>5</sup>**

## 6.2 The Customer Perspective

The introduction of new IT systems into a company is always a change that affects the organization on many levels. An entirely new type of IT system such as web-based applications is potentially even more disruptive. There are many new considerations that didn't exist for conventional types of enterprise software.

Web-based applications have a particularly strong impact on IT departments. The reduced need for infrastructure and internal maintenance can trigger a decrease in IT budgets and staff levels – precisely the main value proposition of most SaaS vendors. It is therefore not surprising that particularly CIOs are fairly critical about SaaS concepts (Alter, 2006).

However, given the increasing adoption of SaaS (see section 5.5), more and more companies will have to come to terms with this new type of IT system. It is therefore important to think early about the influence factors that guide a SaaS implementation.

---

<sup>5</sup> This diagram was inspired by Lorsch and Tierney's Professional Services alignment pyramid (Lorsch and Tierney, 2002)

### **6.2.1 Application Type**

A first major decision is what types of business applications a company can and should buy in the form of web-based software. There are several influence factors for this decision:

- Complexity of required functionality
- Suitability of user situation for a SaaS model (e.g. constant availability of broadband access)
- Confidentiality of the data that will be stored on vendor's infrastructure
- Requirements for availability

These considerations lead many customers to the decision to not use SaaS for truly mission-critical applications (Desisto, Woods and Maoz, 2005), but to concentrate on less sensitive fields. For instance, collaborative software might be important for many internal processes, but it is rarely so critical that a few hours of service interruption would seriously endanger a company's operations.

### **6.2.2 Deployment Scope**

There are different deployment scenarios in which SaaS applications can be used:

- Focused on a departmental deployment: At that level the degree of complexity is lower than for an enterprise-wide rollout. This kind of limited deployment is a good way to get first-hand experience with SaaS applications without the cost and risk associated with an enterprise-wide implementation. Many of the most popular SaaS application types are focused on particular departments (e.g. salesforce automation, HR applications).
- Tactical, rapid deployment: Many SaaS applications are ideal for a quick deployment of limited functionality. A typical example would be a wiki-based collaboration system for a new cross-functional project team.
- Deployment to a formerly underserved user group: The cost-efficiency of SaaS in some cases enables companies to bring IT functionality to user groups that previously did not have access to these means for cost or complexity reasons. For instance, Google targets its Google Apps for Enterprise products specifically to "deskless employees", like factory floor workers, who have limited IT needs but still can benefit from being able to access e-mail, calendars, or a simple word processor (Google, 2007b).
- Enterprise-wide deployment: This is of course the most complex scenario that often triggers further requirements.

It is probably a good idea to first experiment with less risky and limited deployment scenarios before rolling out web-based applications for the whole company.

### **6.2.3 Vendor Type**

Web-based software is sold by a number of different vendor types: startups, experienced SaaS specialists, and traditional software companies that try to get into this new market

segment. As in traditional software, it is important to carefully consider which vendor type to work with. While startups often bring the most innovative functionality, they sometimes lack the support infrastructure for larger deployments. Typically, young companies fail more often than established vendors, and for users of SaaS applications, it is particularly painful when a vendor goes out of business, since service is likely to be interrupted immediately.

It is therefore no surprise that most CIOs still prefer established vendors, even though they might not have the best products (MacManus, 2007). In principle, this is just a case of the classic “credibility gap” that most startups face when selling to enterprise customers (Cusumano, 2004).

#### **6.2.4 Customization and Integration**

The installation, customization and integration of traditional enterprise software is often a major IT project (Vincent et al., 2005). Many IT service firms derive a good portion of their revenues from this type of work.

Web-based applications by their very nature do not have to be installed on customer premises, but they still need to be configured and possibly customized. Although originally SaaS applications had only very limited flexibility for user-specific changes to the system, the leading vendors are increasingly offering advanced customization functionality to match customers’ needs (e.g. Salesforce, 2007). Many vendors partner with external IT service firms who can help software buyers with the customization process.

The integration of a SaaS application with other applications can be difficult, even with modern SOA concepts, as already explained in section 4.3.4. Software buyers therefore need to carefully consider their integration needs and incorporate this factor into their decision for a particular type of software.

#### **6.2.5 Contractual Aspects and Billing Model**

Although most base their pricing on monthly fees per user, vendors use all kinds of contractual models for the actual billing. A few aspects to consider are (Lheureux, 2006):

- Fixed and variable fees (e.g. for storage space)
- Contract duration
- One-time charges for installation and configuration
- Discounts for additional users
- Exit clauses and penalty fees for early termination
- Service level guarantees
- System shutdown periods in case of non-payment (Bona, 2004)

#### **6.2.6 Service and Support Strategy**

Similarly to traditional software, customers of web-based applications need to consider different options for service and support. In most cases, vendors offer support themselves, in some cases as a billable premium service. For some SaaS products, external IT services firms offer third-party support.

A major difference to a traditional software set-up is the fact that in web-based applications, several parties are constantly involved in provisioning the service: The internal IT department (for the operation of the internal network and PCs), multiple Internet providers on the network route between the user and the hosting center, the operator of the hosting infrastructure (which may or may not be identical with the software vendor), and finally the software vendor. This added complexity makes troubleshooting in the case of a system outage significantly more difficult, and users therefore need to consider adequate monitoring measures.

### **6.2.7 Risk Management**

Finally, outsourced web-based applications pose new challenges in risk management. Some of the risk factors that customer organizations have to manage are (Lheureux, 2006):

- Performance and scalability of vendor's infrastructure
- Availability of all the technical components involved
- Security considerations, particularly privacy of hosted data
- Compliance with relevant regulatory standards (e.g. Sarbanes Oxley)
- Disaster recovery plans
- Intellectual property risks
- Vendor viability and appropriate contingency plans

## 7 Summary and Conclusion

This thesis explored the emerging market for web-based enterprise software by analyzing current literature and a sample of 108 companies offering this kind of software product. This final chapter will summarize the results and draw some conclusions.

The market for traditional on-premise enterprise software is in a phase of consolidation. Although the overall market size (including web-based solutions) is growing again after difficult times between 2000 and 2003, consolidation has been the defining trend of the last few years.

Web-based enterprise software (often also referred to as “Software as a service” (SaaS), “on demand” or “hosted applications”) is an important innovation that tries to introduce a new economic and technical model for business applications. The defining characteristics are: a fully web-based user interface, hosted application deployment, a SaaS-based business model (i.e., subscriptions instead of upfront license payments), and the use of a service-oriented architecture for integration.

This study counted 35 major application types that are currently being offered as web-based solutions. Vendors still pursue a large number of different economic and technical strategies for their products, using many combinations of application types, deployment models, and revenue models. A dominant design for any of these dimensions has not emerged yet, with the possible exception of a predominately subscription-based business model.

Web-based enterprise software is clearly an important growth segment in today’s software industry. Almost all of the large vendors of enterprise software are already offering hosted, web-based solutions with Software-as-a-service (SaaS) pricing models. There is also significant entrepreneurial activity, with more than 50% of the companies in the sample examined here founded in or after 2002, with 2005 being the strongest year. Venture capital is still a critically important source of funding for these startups, but there is an equally high share of companies that are bootstrapped, i.e. have not received formal outside investment. This is probably possible due to the comparatively small startup costs for this type of company, but also a reflection of the high number of very young startups in this market.

The SaaS-oriented public software companies, such as RightNow Technologies and Salesforce.com, are experiencing rapid sales growth and attractive market capitalizations. However, these companies currently have a cost structure that is probably typical for an aggressive, rapidly growing startup, but not sustainable over the long run. Still, gross margins for the subscription-based part of their business are attractive, around 85%. As with traditional software companies, the mix between product-based and professional-service based revenues is a major strategic consideration that can strongly influence overall margins.

The data about customer adoption of SaaS are not very clear and reliable. Most studies suggest that in 2006, web-based CRM accounted for slightly more than 5% of the total CRM market, with other application types below this level. However, customer awareness of SaaS and the willingness to adopt this new type of software seems to be signifi-

cantly greater than that and is growing very quickly. This pattern is probably typical for a rapidly emerging technology.

Chapter 6 of this thesis tried to outline strategic frameworks for both the vendors and users of web-based applications, based on the insights gained from the previous chapters. Vendors have to make a number of specific strategic decisions that range from more generic dimensions, such as which segment of target customers to address, to very SaaS-specific subjects, such as the right application hosting strategy. For customers, the decision for web-based applications is not entirely straightforward, since there are several organizational, technical and risk factors that are specific to this type of IT system and should be carefully considered.

Finally, the central research question of this thesis can't be answered unambiguously from the data available. The analysis demonstrated that web-based enterprise software is a new business model that is clearly attractive for certain segments of vendors and customers. It also showed that this type of software is a major trend in the software industry today, but a trend that is still in its early stages, probably not even accounting for 5% of total revenues from enterprise software and less than 1% of the global software market overall. However, revenues for vendors in this field are growing rapidly, customer awareness is increasing strongly, and there is very significant entrepreneurial activity. Therefore, there is reason to believe that web-based software has passed an inflection point on the typical S-curve that radical technological innovations tend to follow (Utterback, 1994). Web-based enterprise software exhibits most, but not unequivocally all characteristics of a new-market disruption (Christensen, 2003) that particularly targets current non-user of the established technology generation.

It is too early, however, to predict winning strategies or even the type of company that will be successful in this wave of innovation. The market is apparently still experimenting with many different designs, and only time will tell which combinations of target markets, business models and technologies will be the most successful.

## 8 Appendix

List of companies analyzed in the sample:

Vendor Name	Country/ US State	Found ed	Application Type	Revenue Mod- els	Deployment Models	Primary source of capital
24SevenOffice	Nor- way	1997	Calendar, CRM	Subscription	Remotely hosted service	Boot- strapped
30 Boxes	CA	2005	Calendar	Free	Remotely hosted service	Boot- strapped
37signals	IL	1999	Project manage- ment, Calendar, Team collabora- tion	Subscription, Free	Remotely hosted service	VC
actindo	Ger- many	1998	Content Man- agement, Finance	Subscription	Remotely hosted service	Boot- strapped
Adaptive Plan- ning	CA	2003	BPM	Subscription	Remotely hosted service, Local instal- lation	VC
Adobe	CA	1982	Application De- velopment, Imag- ing	Upfront license, Subscription	Remotely hosted service, Local instal- lation	Public
Amazon Web- services	WA	1994	Infrastructure	Pay per use, Subscription	Remotely hosted service	Public
Approver.com	CA	2006	Workflow	Subscription	Remotely hosted service	Cross- subsidized
Astea Interna- tional	PA	1979	Field Service Management	Upfront license	Remotely hosted service, Local instal- lation	Public
AutoTask	NY	2001	Project manage- ment, Profes- sional Services Automation, CRM	Subscription	Remotely hosted service	VC
BlogTronix	CA	2005	Blog	Upfront license, Free, Subscrip- tion	Remotely hosted service, Local instal- lation	Boot- strapped
Blueroads	CA	2001	CRM, Team col- laboration	Subscription	Remotely hosted service	VC
Bottomline Technologies	NH	1989	Finance	Upfront license, Professional Services	Remotely hosted service, Local instal- lation	Public
BrainKeeper	VA	2006	Wiki	Subscription	Remotely hosted service	Boot- strapped
Brainloop	Ger- many	2000	Team collabora- tion, Storage	Upfront license, Subscription	Remotely hosted service, Local instal- lation	Boot- strapped

Business Objects / crystalreports.com	France	1990	Business Intelligence	Upfront license, Subscription	Remotely hosted service, Local installation	Public
CAS	Germany	1986	CRM	Upfront license, Professional Services, Subscription	Remotely hosted service, Local installation	Bootstrapped
Caspio	CA	2000	Application Development	Upfront license, Free, Professional Services, Subscription	Remotely hosted service, Local installation	Bootstrapped
Celum Imagine	Austria	1999	Imaging	Upfront license, Subscription	Remotely hosted service, Local installation	Bootstrapped
Centive	MA	1997	Salesforce management	Professional Services, Subscription	Remotely hosted service	VC
Central Desktop	CA	2004	Calendar, Team collaboration, Conferencing	Free, Subscription	Remotely hosted service	Bootstrapped
cierp3 / Intraprend	Germany	1982	HR, ERP, CRM, Finance	Upfront license, Subscription	Remotely hosted service, Local installation	Bootstrapped
Cogenz	United Kingdom	2005	Team collaboration, Content Management	Subscription	Remotely hosted service	VC
CollectiveX	MD	2006	Social Networking, Team collaboration	Advertising, Subscription	Remotely hosted service	Bootstrapped
Collmex	Germany	2003	ERP, Finance	Subscription	Remotely hosted service	Bootstrapped
ConceptShare	Canada	2005	Imaging, Team collaboration	Free, Subscription	Hardware appliance, Remotely hosted service, Local installation	Bootstrapped
Concur Technologies	WA	1993	Finance	Subscription	Remotely hosted service	Public
Dabble DB	Canada	2005	Application Development, Database	Subscription	Remotely hosted service	VC
DropSend	United Kingdom	2004	Storage, Team collaboration	Subscription	Remotely hosted service	Cross-subsidized
EchoSign	CA	2005	Workflow	Free, Subscription	Remotely hosted service	VC
EditGrid (TnC)	Hong Kong	2003	Spreadsheet	Subscription, Free	Remotely hosted service	Cross-subsidized
Endo7 / Group-e	Italy	2001	Team collaboration	Subscription	Remotely hosted service, Local installation	Cross-subsidized
Entellium	WA	2000	CRM	Subscription	Remotely hosted service	VC



Etelos	WA	1999	Salesforce management, Project management, Application Development, Content Management	Upfront license, Advertising, Subscription	Hardware appliance, Virtual appliance, Remotely hosted service, Local installation	Bootstrapped
Foldera	CA	2001	Calendar, Search, Team collaboration, Content Management	Free	Remotely hosted service	Public
ForeSoft	IL	1999	Application Development, CRM	Subscription	Remotely hosted service	Bootstrapped
FreshBooks	Canada	2002	Finance	Free, Subscription	Remotely hosted service	Bootstrapped
Fujitsu/Glovia	CA	1990	ERP, Finance, CRM, Business Intelligence	Upfront license, Subscription	Remotely hosted service, Local installation	Public
Genius.com	CA	2004	CRM, Salesforce management	Subscription	Remotely hosted service	VC
gOffice	CA	2003	Word Processing	Subscription	Remotely hosted service	Bootstrapped
Google	CA	1998	Word Processing, e-Mail, Calendar, Wiki, Portal, Search, Team collaboration, Conferencing	Upfront license, Advertising, Subscription	Remotely hosted service, Hardware appliance	Public
Goplan	Portugal	2006	Project management, Calendar, Team collaboration, Conferencing	Free, Subscription	Remotely hosted service	Cross-subsidized

IBM	NY	1911	e-Mail, Calendar, Content Management, Mobile Access, Application Development, Team collaboration, Conferencing	Upfront license, Professional Services, Subscription	Hardware appliance, Remotely hosted service, Local installation	Public
Idee	Canada	1999	Imaging	Upfront license, Advertising	Remotely hosted service, Local installation	VC
iNetOffice	WA	2004	Word Processing	Free	Remotely hosted service	Bootstrapped
Intacct	CA	1999	Finance	Subscription	Remotely hosted service	VC
Intensil	CA	2001	Project management, Team collaboration, Wiki	Subscription	Remotely hosted service	Bootstrapped
Intuit	CA	1983	Database, Finance, Application Development	Upfront license, Subscription	Remotely hosted service, Local installation	Public
iUpload	Canada	1998	Blog, Content Management	Subscription	Remotely hosted service	Bootstrapped
JasperSoft	CA	2006	Business Intelligence	Upfront license, Professional Services, Subscription	Remotely hosted service, Local installation	VC
Joyent	CA	2005	Storage, Calendar, e-Mail, Team collaboration, Application Development	Subscription	Remotely hosted service	Bootstrapped
Koral	CA	2006	Content Management	Free, Subscription	Remotely hosted service	VC
LucidEra	CA	2005	Business Intelligence	Subscription	Remotely hosted service	VC
Microsoft	WA	1975	e-Mail, Project management, Calendar, Mobile Access, Team collaboration, CRM	Upfront license, Advertising, Free, Subscription	Remotely hosted service, Local installation	Public
My Payment	CA	2005	Finance	Pay per use,	Remotely hosted	VC

Network				Subscription	service	
myfactory	Germany	2002	HR, ERP, CRM	Upfront license, Subscription	Remotely hosted service, Local installation	Bootstrapped
Near-Time	NC	2006	Project management, Calendar, Team collaboration, Wiki, Blog	Subscription	Remotely hosted service	VC
Netsuite	CA	1998	ERP, CRM, Finance	Professional Services, Subscription	Remotely hosted service	VC
Netvibes	France	2005	Portal	Advertising, Free	Remotely hosted service	VC
NewsGator	CO	2003	RSS, Portal	Upfront license, Subscription	Remotely hosted service, Local installation	VC
NumSum	CA	2005	Spreadsheet	Free	Remotely hosted service	Bootstrapped
OmniDrive	CA	2004	Storage	Free, Subscription	Remotely hosted service, Local installation	Bootstrapped
OpenAir	MA	1999	Professional Services Automation	Subscription	Remotely hosted service	VC
Open-Xchange	NY / Germany	2005	Team collaboration	Professional Services, Subscription	Remotely hosted service, Local installation	Bootstrapped
OpSource	CA	2002	Application Development, Infrastructure	Professional Services, Pay per use, Subscription	Remotely hosted service	VC
Oracle	CA	1977	HR, Salesforce management, Calendar, ERP, Finance, Business Intelligence, Content Management, Portal, Application Development, CRM	Upfront license, Professional Services, Subscription	Remotely hosted service, Local installation	Public
Panthius	Ireland	2004	ERP, Supply Chain, CRM, Business Intelligence	Professional Services, Subscription	Remotely hosted service	Bootstrapped
Parature	VA	1999	CRM	Advertising, Subscription	Remotely hosted service	VC

Plateau	VA	1996	HR, e-Learning	Upfront license, Professional Services, Subscription	Remotely hosted service, Local installation	VC
Project 360	CO	2004	Team collaboration, Project management	Free, Subscription	Remotely hosted service	Bootstrapped
Purkinje	MO	1995	Healthcare, ERP	Pay per use, Subscription	Remotely hosted service	VC
RightNow	MT	1997	CRM	Upfront license, Professional Services, Subscription	Remotely hosted service, Local installation	Public
rPath	NC	2005	Application Development, Appliance	Upfront license, Subscription	Remotely hosted service, Local installation	VC
Sage	United Kingdom	1981	HR, ERP, Finance, CRM	Upfront license, Subscription	Remotely hosted service, Local installation	Public
Salary.com	MA	1999	HR	Pay per use, Subscription	Remotely hosted service	Public
salesforce.com	CA	1999	CRM, Salesforce management	Subscription	Remotely hosted service, Local installation	Public
SAP	Germany	1972	ERP, Supply Chain, CRM, Finance	Upfront license, Professional Services, Subscription	Remotely hosted service, Local installation	Public
ScanR	CA	2005	Imaging, Team collaboration	Free	Remotely hosted service	VC
Serebrum	NJ	2000	Application Development, Finance	Professional Services	Remotely hosted service	Bootstrapped
ShareMethods	NJ	2003	Team collaboration, Content Management	Subscription	Remotely hosted service	Bootstrapped
Simulat	CA	2003	Imaging, e-Learning, Conferencing	Advertising, Professional Services, Subscription	Remotely hosted service	Bootstrapped
Siterra	CA	2001	Business Intelligence, Project management	Upfront license, Subscription	Remotely hosted service, Local installation	VC
Skip 5	Switzerland	2005	ERP, Finance, CRM	Subscription	Remotely hosted service	Bootstrapped
SlideShare	CA	2006	Presentation, Team collaboration	Free	Remotely hosted service	Bootstrapped

Smartsheet.com	WA	2005	Spreadsheet, Project management	Subscription, Free	Remotely hosted service	Bootstrapped
SocialText	CA	2002	Wiki	Upfront license, Free, Subscription	Hardware appliance, Virtual appliance, Remotely hosted service, Local installation	VC
Software Europe Ltd.	United Kingdom	1989	Infrastructure, Finance	Upfront license, Subscription	Remotely hosted service, Local installation	Bootstrapped
Solodox (NetBeans)	Japan	1994	Word Processing	Free	Remotely hosted service	Bootstrapped
Spresent	TX	2006	Presentation	Free	Remotely hosted service	Bootstrapped
SuccessFactors	CA	2001	HR	Subscription	Remotely hosted service	VC
SugarCRM	CA	2004	CRM	Upfront license, Subscription	Hardware appliance, Virtual appliance, Remotely hosted service, Local installation	VC
Taleo	CA	1999	HR	Professional Services, Subscription	Remotely hosted service	Public
Teqlo	CA	2005	Application Development	Free	Remotely hosted service	VC
Thinkature	MA	2006	Conferencing	Free	Remotely hosted service	Bootstrapped
ThinkFree	Korea	1990	Presentation, Word Processing, Spreadsheet	Upfront license, Subscription	Remotely hosted service, Local installation	VC
Time Search	Canada	2005	Calendar	Free	Remotely hosted service	Bootstrapped
Trovix	CA	2002	Search, HR	Subscription	Remotely hosted service	VC
Trumba	WA	2003	Calendar	Subscription	Remotely hosted service	VC
Ultimate Software	FL	1990	HR	Upfront license, Subscription	Remotely hosted service, Local installation	Public
Umantis	Switzerland	2000	HR	Upfront license	Remotely hosted service, Local installation	
Vertabase	MI	1999	Project management	Upfront license, Subscription	Remotely hosted service, Local installation	Cross-subsidized
WebEx	CA	1996	Conferencing	Pay per use, Subscription	Remotely hosted service	Public
Workday	CA	2005	ERP, CRM, Finance	Subscription	Remotely hosted service	Bootstrapped
Wufoo	FL	2005	Application Development	Subscription	Remotely hosted service	VC
Xactly	CA	2005	Salesforce management	Subscription	Remotely hosted service	VC

Xcellery (Collaborall)	CA	2006	Spreadsheet	Free	Remotely hosted service	Bootstrapped
YouSendIt	CA	2003	Storage, Team collaboration	Free, Subscription	Remotely hosted service	VC
Zoho / Advent-Net	CA	1996	Presentation, Word Processing, Spreadsheet, Project management, Calendar, Database, Wiki, Infrastructure, Application Development, Conferencing, CRM	Upfront license, Free, Subscription	Remotely hosted service, Local installation	Cross-subsidized

## 9 References

- Adobe Systems. 2006. Annual Report, Form 10-K. EDGAR, [http://www.sec.gov/Archives/edgar/data/796343/000110465907007286/a07-3000\\_110k.htm](http://www.sec.gov/Archives/edgar/data/796343/000110465907007286/a07-3000_110k.htm), downloaded April 3, 2007.
- AdventNet, Inc. 2007. Corporate website. <http://www.adventnet.com>, accessed March 25, 2007.
- Agassi, Shai. 2006. Business Process Co-Innovation, "Enterprise 3.0". Presentation at Software 2006 conference, Santa Clara, April 4, 2006. [www.sandhill.com/conferences/sw2006\\_materials/Shai\\_Agassi\\_SW2006\\_final.pdf](http://www.sandhill.com/conferences/sw2006_materials/Shai_Agassi_SW2006_final.pdf), downloaded March 1, 2007.
- Alarm:clock. 2007. Alarm:clock blog. <http://www.thealarmclock.com>.
- Alter, Allan E. 2006. Are CIOs Ignoring Web 2.0 Technologies? CIO Insight, May 10, 2006. <http://www.cioinsight.com/article2/0,1540,1960180,00.asp>, downloaded March 6, 2007.
- Amazon Web Services. 2007. Amazon web services website. <http://aws.amazon.com>, accessed March 30, 2007.
- AMR Research. 2006. Enterprise Resource Planning Software Will Grow to \$29 Billion in 2006. Press Release, October 10, 2006.
- Aral, Sinan, Erik Brynjolfsson and Marshall W. Van Alstyne. 2006. Information, Technology and Information Worker Productivity: Task Level Evidence. Working Paper, via SSRN, <http://ssrn.com/abstract=942310>, downloaded March 30, 2007.
- Autodesk. 2006. Annual Report, Form 10-k. EDGAR, <http://www.sec.gov/Archives/edgar/data/769397/000119312506068809/d10k.htm>, downloaded April 3, 2007.
- Babcock, Charles. 2006. Selling Software Apps Like Refrigerators, OS and All. InformationWeek, November 13, 2006, Issue 1114, via ProQuest.
- Band, William. 2006. The ABCs of CRM Success. Optimize Magazine, January 2006, Issue 51. <http://www.optimize.com/article/showArticle.jhtml?articleId=175700405>, downloaded March 31, 2007.
- Bank, David. 2004. Oracle Reports Earning Gains of 11%. The Wall Street Journal, March 12, 2004, page A3, via ProQuest.
- Bednarz, Ann. 2006. Hosted apps handle holiday rush. Network World, January 9, 2006.
- Berryman, Ken, Joel Jones, James Manyika, and MR Rangaswami. 2006. Software 2006 Industry Report. SandHill.com, [http://www.sandhill.com/conferences/sw2006\\_materials/SW2006\\_Industry\\_Report.pdf](http://www.sandhill.com/conferences/sw2006_materials/SW2006_Industry_Report.pdf), downloaded March 10, 2007.
- Blau, John. 2007. SAP demonstrates A1S to select groups. InfoWorld online, March 15, 2007, [http://www.infoworld.com/article/07/03/15/HNsapdemonstratesa1s\\_1.html](http://www.infoworld.com/article/07/03/15/HNsapdemonstratesa1s_1.html), downloaded March 31, 2007.

- Blogtronix, LLC. 2007. Corporate website. <http://www.blogtronix.com>, accessed March 18, 2007.
- Bona, Alex. 2004. CRM on demand is more expensive than you think. Stamford: Gartner, Inc. May 19, 2004.
- Bokhari, Zaineb. 2006. Standard & Poor's Industry Surveys, Computers: Software. New York: Standard & Poor's, October 19, 2006.
- Briody, Dan. 1999. Picking the PC's new directions. InfoWorld, June 21, 1999, Issue 25, via ProQuest.
- Bulkeley, William M. 2004. IBM Strikes Deal With Rival Lenovo. The Wall Street Journal, December 8, 2004, via ProQuest.
- Campbell-Kelly, Martin. 2004. From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry. Cambridge, MA: The MIT Press.
- Carr, Nicholas. 2006. SaaS adoption set to explode. Rough Type blog, November 27, 2006. [http://www.routhtype.com/archives/2006/11/cio\\_interest\\_in.php](http://www.routhtype.com/archives/2006/11/cio_interest_in.php), downloaded January 21, 2007.
- Chou, Timothy. 2005. The End of Software: Transforming Your Business for the On Demand Future. Indianapolis: Sams Publishing.
- Christensen, Clayton M. 1997. The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail. Boston: Harvard Business School Press.
- Christensen, Clayton M., and Michael E. Raynor. 2003. The Innovator's Solution : Creating and Sustaining Successful Growth. Boston: Harvard Business School Press.
- Cook, John. 2006. Hot startup unveils plan. Seattle Post-Intelligencer, August 4, 2006. [http://seattlepi.nwsourc.com/business/280063\\_smartsheet04.html](http://seattlepi.nwsourc.com/business/280063_smartsheet04.html), downloaded March 27, 2007.
- CRN magazine. 2004. Software for Hire. November 2004.
- Cusumano, Michael A. 2004. The Business of Software : What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad. New York: Free Press.
- Da Silva, Dominic. 2006. Ruby on Rails: All Aboard the Fast Train to Web Application Development. Developer.com, December 7, 2006, [www.developer.com/db/article.php/3647796](http://www.developer.com/db/article.php/3647796), downloaded March 31, 2007.
- Davenport, Thomas H. 1998. Putting the enterprise into the enterprise system. Harvard Business Review 76, (4) (July - August 1998): 121.
- Delaney, Kevin J. and Robert A. Guth. 2006. Google's Free Web Services Will Vie With Microsoft Office. The Wall Street Journal, October 11, 2006, p B1, via wsj.com.
- Delaney, Kevin J. 2006. Google Buys JotSpot to Expand Online Document-Sharing Service. The Wall Street Journal, November 1, 2006, via ProQuest.
- Desisto, Robert P., Ben Pring, Benoit J. Lheureux, and Frances Karamouzis. 2006. SaaS delivery challenges on-premise software. Stamford: Gartner, Inc. September 26, 2006.



Desisto, Robert P., Jeff Woods, and Michael Maoz. 2005. On-demand services won't dominate the future of business applications. Stamford: Gartner, Inc., September 29, 2005.

Etelos, Inc. 2007. Corporate website. <http://www.etelos.com>, accessed March 30, 2007.

Evans, David S., Andrei Haigu and Richard Schmalensee. 2006. Invisible Engines: How Software Platforms Drive Innovation and Transform Businesses. Cambridge, MA: The MIT Press.

Economist. 2006. Spot the dinosaur: Microsoft's core business is under threat from online software. The Economist, March 30, 2006, via Economist.com.

Forbes Magazine. 2005. The Forbes Global 2000. <http://www.forbes.com/2005/03/30/05f2000land.html>, downloaded March 31, 2007.

Foremski, Tom. 2001. Software groups take a hard knock over grim outlook. Financial Times, March 3, 2001, via ProQuest.

Gartner Dataquest. 2006. Market Share: CRM Software Worldwide, 2005. Stamford: Gartner Inc., June 2006, via eMarketer.

Gawer, Annabelle, and Michael A. Cusumano. 2002. Platform leadership : How intel, microsoft, and cisco drive industry innovation. Boston: Harvard Business School Press.

Google. 2006. Annual Report, Form 10-K. EDGAR, <http://www.sec.gov/Archives/edgar/data/1288776/000119312507044494/0001193125-07-044494-index.htm>, downloaded April 3, 2007.

Google. 2007a. Google Introduces New Business Version of Popular Hosted Applications. Press Release, February 22, 2007. [http://www.google.com/intl/en/press/pressrel/google\\_apps.html](http://www.google.com/intl/en/press/pressrel/google_apps.html), downloaded March 31, 2007.

Google. 2007b. Google Apps website. <http://www.google.com/a/>, accessed March 29, 2007.

Graham, Jefferson. 2006. Adobe Acrobat reader upgrades, learns new tricks. USA Today, September 18, 2006, via usatoday.com.

Graham, Paul. 2005. Web 2.0. <http://www.paulgraham.com/web20.html>, downloaded January 21, 2007.

Hax, Arnolde and Dean Wilde. 2001. The Delta Project: Discovering New Sources of Profitability in a Networked Economy. Houndmills: Palgrave.

Hein, Bettina. 2007. 0 + 0 = 1: The Appliance Model of Selling Software Bundled with Hardware. Master's thesis at the Massachusetts Institute of Technology, Cambridge, MA.

Hof, Rob. 2006. 37Signals, 1 Big New Investor: Jeff Bezos. BusinessWeek Online, July 20, 2006. [http://www.businessweek.com/the\\_thread/techbeat/archives/2006/07/37signals\\_1\\_big.html](http://www.businessweek.com/the_thread/techbeat/archives/2006/07/37signals_1_big.html), downloaded March 31, 2007.

Hoover, J. Nicholas. 2007. Microsoft's Next Move in ERP. Information Week, March 12, Issue 1127, via ProQuest.

12 Technologies. 2006. Annual Report, Form 10-K. EDGAR, <http://www.sec.gov/Archives/edgar/data/1009304/000119312507070753/0001193125-07-070753-index.htm>, downloaded April 3, 2007.

IBM Corporation. 2006. Annual Report, Form 10-K. EDGAR, <http://www.sec.gov/Archives/edgar/data/51143/000104746907001434/a2175501z10-k.htm>, downloaded March 30, 2007.

IETF. 2007. The IETF Standards Process. <http://www.ietf.org/IETF-Standards-Process.html>, downloaded April 3, 2007.

Inc. magazine. 2004. 2004 Inc. 500: Top 25 Companies. <http://www.inc.com/resources/inc500/2005/profiles/027.html>, accessed March 24, 2007.

Infor. 2007. Corporate website, <http://www.infor.com>, accessed March 25, 2007.

Intuit, Inc. 2007. Corporate Website. <http://www.intuit.com>, accessed February 23, 2007.

Junnarkar, Sandeep. 2003. IBM talks up 'computing on demand'. CNet News.com, [http://news.com.com/IBM+talks+up+computing+on+demand/2100-1001\\_3-963807.html](http://news.com.com/IBM+talks+up+computing+on+demand/2100-1001_3-963807.html), downloaded March 14, 2007.

Lacy, Sarah. 2006. Net Suite Gets Ready For Its Close-Up. BusinessWeek, December 11, 2006, via BusinessWeek Online, accessed March 24, 2007.

LaMonica, Martin. 2006. No Microsoft 'Web Office' on tap, says Ozzie. CNet News.com, October 27, 2006. [http://news.com.com/2061-12572\\_3-6130134.html](http://news.com.com/2061-12572_3-6130134.html), downloaded January 21, 2007.

Lawson Software. 2006. Annual Report, Form 10-K. EDGAR, <http://www.sec.gov/Archives/edgar/data/1344632/000110465906058022/0001104659-06-058022-index.htm>, downloaded April 3, 2007.

Le Clair, Craig. 2005. How to succeed in the enterprise software market. Hershey, PA: CyberTech Publishing.

Lheureux, Benoit J., Robert P. Desisto and Michael Maoz. 2006. Evaluating Software-as-a-Service Providers: Questions to Ask Potential SaaS Providers. Stamford: Gartner, Inc., April 18, 2006.

Lorsch, Jay W. and Thomas J. Tierney. 2002. Aligning the Stars. Boston: Harvard Business School Press.

MacManus, Richard. 2007. CIOs Spurn Web 2.0 Startups – Enterprises Want Suites and Large, Incumbent Software Vendors. Read/Write Web blog, March 21, 2007. [www.readwriteweb.com/archives/cios\\_spurn\\_web2\\_startups.php](http://www.readwriteweb.com/archives/cios_spurn_web2_startups.php), downloaded March 22, 2007.

Maoz, Michael. 2006. CRM Beyond Web 2.0 Features 'Rich Clients' and Software as a Service. Stamford: Gartner, Inc., January 17, 2006.

Maoz, Michael, and Robert P. Desisto. 2006. When Will the SaaS Model Support Complex Business Processes? Stamford: Gartner, Inc., July 6, 2006.

McAfee, Andrew. 2006. Mastering the three worlds of information technology. Harvard Business Review, 84, November 2006, page 141.

McDougall. 2005. IBM-Corio Deal Makes ASPs Ripe for Picking. Information Week, January 31, 2005.  
<http://www.informationweek.com/showArticle.jhtml?articleID=59100480>, downloaded April 3, 2007.

McGeary, Zachary, Vikram Shegal, and David Daniels. 2005. Hosted CRM: Evaluating Market Opportunity for Customer Service Technologies. New York: JupiterResearch, September 28, 2005.

Messerschmitt, David G. 1999. Understanding Networked Applications. San Francisco: Morgan Kaufmann.

Microsoft Corporation. 2005. Microsoft Previews New Windows Live and Office Live Services. Press Release, November 1, 2005,  
<http://www.microsoft.com/presspass/press/2005/nov05/11-01PreviewSoftwareBasedPR.mspx>, downloaded March 31, 2007.

Microsoft Corporation. 2007. Fast Facts About Microsoft.  
[http://www.microsoft.com/presspass/inside\\_ms.mspx](http://www.microsoft.com/presspass/inside_ms.mspx), downloaded March 31, 2007.

Moore, Geoffrey. 2002. Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers. Revised edition. New York: Collins.

Moore, John. 2005. ERP Makes a Comeback. The Channel Insider, August 2005, via LookSmart.

NVCA. 2006. National Venture Capital Association Industry Statistics.  
<http://www.nvca.org/ffax.html>, downloaded March 15, 2007.

Oracle Corporation. 2006. Annual Report 2006, Form 10-K. EDGAR,  
<http://www.sec.gov/Archives/edgar/data/1341439/000119312506151154/0001193125-06-151154-index.htm>, downloaded March 31, 2007.

Oracle. 2007. Corporate Website, <http://www.oracle.com>, accessed February 20, 2007.

O'Reilly, Tim. 2005. What is Web 2.0. Design patterns and business models for the next generation of software. Sebastopol: O'Reilly Media,  
<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>, downloaded February 02, 2007.

O'Reilly, Tim. 2006. Web 2.0 Compact Definition: Trying Again. Sebastopol: O'Reilly Media, [http://radar.oreilly.com/archives/2006/12/web\\_20\\_compact.html](http://radar.oreilly.com/archives/2006/12/web_20_compact.html), downloaded March 22, 2007.

Ozzie, Ray. 2005. The Internet Services Disruption. Internal Microsoft corporation memo, published on CNet News.com, [http://news.com.com/2102-1016\\_3-5942232.html](http://news.com.com/2102-1016_3-5942232.html), downloaded March 30, 2007.

Porter, Michael. 1980. Competitive Strategy: Techniques for Analyzing Industries and Competitors. New York: Free Press.

Prentice, Stephen. 2007. From Computers to the Appliance Model: Is 2007 the Tipping Point? Stamford: Gartner, Inc., March 14, 2007.

Pring, Ben. 2005. Adoption of software as a service is happening outside of CRM. Stamford: Gartner, Inc., December 16, 2005.

Radicati Group, Inc. 2006. Microsoft Exchange Market Share Statistics, 2006. Palo Alto: Radicati Group, March 2006.

RightNow Technologies. 2006. Annual Report, Form 10-K. EDGAR, <http://www.sec.gov/Archives/edgar/data/1111247/000110465907019100/0001104659-07-019100-index.htm>, downloaded March 24, 2007.

RightNow Technologies. 2007. Corporate website, <http://www.rightnow.com>, accessed March 17, 2007.

Rosenberg, Barry and Craig Wright. 2007. Software As A Service: Value To Go -- SaaS can deliver great long-term value. Optimize magazine, March 2007, Issue 65, via Pro-Quest.

Sage Group Plc. 2007. Corporate Website. <http://www.sage.com>, accessed March 10, 2007.

Salesforce.com. 2006. Annual Report, Form 10-K. EDGAR, <http://www.sec.gov/Archives/edgar/data/1108524/000119312507051045/d10k.htm>, downloaded March 30, 2007.

Salesforce.com. 2007a. Corporate website. <http://www.salesforce.com>, accessed March 23, 2007.

Salesforce.com. 2007b. AppExchange website. <http://www.appexchange.com>, accessed March 23, 2007.

SAP AG. 2005. Annual Report 2005, via [www.sap.com](http://www.sap.com).

SAP AG. 2006. Annual Report 2006, via [www.sap.com](http://www.sap.com).

SAP AG. 2007. Corporate Website, <http://www.sap.com>, accessed February 20, 2007.

Shankland, Steven. 2006. Windows bumps Unix as top server OS. CNet News.com, February 21, 2006. [http://news.com.com/2100-1016\\_3-6041804.html](http://news.com.com/2100-1016_3-6041804.html), downloaded April 1, 2007.

Shapiro, Carl and Hal R. Varian. 1998. Information Rules: A Strategic Guide to the Network Economy. Boston: Harvard Business School Press.

Siebel Systems. 2004. Annual Report, Form 10-K. EDGAR, <http://www.sec.gov/Archives/edgar/data/1006835/000100683505000012/form10k.htm>, downloaded April 1, 2007.

Sims, David. 2007. Gartner Finds SaaS Market to Hit \$19.3 Billion by 2011. TMCnet, March 7, 2007. [www.tmcnet.com/usubmit/2007/03/07/2398768.htm](http://www.tmcnet.com/usubmit/2007/03/07/2398768.htm), downloaded March 20, 2007.

Smith, Gary E. 2007a. One Year Later: SaaS Breaks Out – Saugatuck Research. SAAS Network Architect, March 8, 2007. <http://saasnetworkarchitect.com/2007/03/08/one-year-later-saas-breaks-out--saugatuck-research.aspx>, downloaded March 31, 2007.

Smith, Gary E. 2007b. SaaS Adoption: Skyrocketing on all cylinders! – Saugatuck Research. SAAS Network Architect, March 29, 2007.  
<http://saasnetworkarchitect.com/2007/03/29/saas-adoption-skyrocketing-on-all-cylinders--saugatuck-technology.aspx>, downloaded March 31, 2007.

Snyder, William R. 2006. The software value gap is driving a wedge between vendors and customers. Stamford: Gartner, Inc. December 22, 2006.

TechCrunch. 2007. TechCrunch blog, <http://www.techcrunch.com>.

ThinkFree, Inc. 2007. Corporate website, <http://www.thinkfree.com>, accessed March 20, 2007.

Thurrott, Paul. 2003. OS Market Share: Microsoft Stomps the Competition. Windows ITPro, October 9, 2003.  
<http://www.windowsitpro.com/Article/ArticleID/40481/40481.html>, downloaded April 1, 2007.

Turner, Rob. 2005. Larry's Kids. Inc Magazine, February 2005.

Utterback, James M. 1994. Mastering the Dynamics of Innovation: How Companies Can Seize Opportunities in the Face of Technological Change. Boston: Harvard Business School Press.

VentureBeat. 2007. VentureBeat blog, <http://www.venturebeat.com>.

Vincent, Philippe, Tim Jellison, Bart Hughes, and Philipp Stauffer. 2005. The Future of Enterprise Software: How Software Companies can Achieve High Performance in an Era of Disruptive Change and Uncertainty. Hamilton, Bermuda: Accenture.  
[http://www.accenture.com/Global/Services/By\\_Industry/Electronics\\_and\\_High\\_Tech/R\\_and\\_I/enterprise\\_software\\_pov.htm](http://www.accenture.com/Global/Services/By_Industry/Electronics_and_High_Tech/R_and_I/enterprise_software_pov.htm), downloaded March 30, 2007.

W3C. 2007. World Wide Web Consortium web site. <http://www.w3.org>, accessed March 30, 2007.

Waxer, Cindy. 2006. The Fight Over CRM. Chief Executive, Jan/Feb 2006, Issue 215, via ProQuest.

Waters, Richard. 2006. Full transcript: FT interview with Larry Ellison. Financial Times, April 18, 2006, via FT.com.

White, Bobby and Roger Cheng. 2007. Cisco to Acquire Webex. The Wallstreet Journal, March 16, 2007, page B4, via wsj.com.

Wiesmann, Gerrit. 2007. SAP eager for new product launch. Financial Times, March 16, 2007.

Workday, Inc. 2007. Corporate website. <http://www.workday.com>, accessed March 15, 2007.

Xactly. 2007. Corporate website. <http://www.xactly.com>, accessed March 27, 2007.

Yahoo Finance. 2007. Application Software Company List.  
<http://biz.yahoo.com/p/821mkt.html>, downloaded March 14, 2007.