Eco-Bridge LLC

CONSULTING

www.eco-bridge.com

A disruptive new software paradigm is changing the economics of the software industry. Businesses must understand this new model to leverage it. Incumbent software vendors must review their risk appetites and invest in it, or risk losing market share.

Software As A Service:

Why It Matters

By Anand Divekar

August 2007

Table of Contents

Executive Summary

Customer spending in the software industry has been impacted in the last few years by several crises driving sharply reduced demand for high priced software. A major concern is the annual cost of owning and managing software applications, which can be up to four times the cost of the initial purchase. As a result, companies end up spending more than 75% of their IT budgets just on maintaining and running existing systems and software infrastructure.

Software as a Service (SaaS) represents a paradigm shift in the way software is built, delivered, operated, maintained and upgraded. It leverages the internet for its many benefits and transfers responsibility for software ownership, maintenance, upgrades and operations to the vendor. SaaS also represents a paradigm shift for end users as regards how they will buy, use and pay for software in the future.

Market penetration is accelerating across various verticals and software categories. SaaS offers a range of value propositions including Speed to Value, Total Cost of Ownership and vendor accountability, but also has limitations based on current architectures, integration frameworks and standards. IT departments supporting SaaS deployments should consider using best practices such as the eSCM.

The ecosystem is evolving, with startups embracing the SaaS model, vendors evaluating options for migrating existing products and making datacenter investments, and changes being negotiated to the channel partner model. Investors should take note of the enhanced growth prospects and valuations of vendors who migrate to SaaS.

We believe this report will provide valuable insights to software companies and their customers, as well as investors looking to invest in the next disruptive technology trend.

Continued...

Market Data and Analyst Predictions

SaaS is increasingly gathering momentum in the software industry as can be seen from the following metrics:

- The worldwide SaaS market reached $6.3 billion in 2006 and is forecast to grow to $19.3 billion by year-end 2011, according to Gartner, Inc.
- Per IDC, 10% of the market for enterprise software will migrate to a pure SaaS model by 2009.[1]
- Credit Suisse's March 11, 2007 "On Demand Market Forecast" projects "On Demand software to grow at a CAGR of 36% to roughly $21 billion in 2011 from over $4 billion in 2006."
- A McKinsey survey in the fourth quarter of 2006 found that 61% of North American companies with sales over $1 billion plan to adopt one or more SaaS applications in 2007, a dramatic increase from the 38% planning to install SaaS apps in 2005.
- A Saugatuck survey indicated the number of companies over $1 billion in revenue that said they were planning to deploy SaaS for mission critical applications more than quadrupled over last year, from 13% to 53%.[2]

Conclusions

Businesses, vendors and investors can no longer ignore SaaS trends and their financial implications. The ecosystem of the software industry and the maturity of associated technologies will evolve dramatically in coming years, driving changes to the way businesses and users perceive software, its purchase and consumption, usage, performance, value and billing as well as associated data management practices.

These changes could ripple through to the IT labor industry (shrinking customer demand for operations staff & systems integrators) and the hardware industry (growing demand for data center related hardware and software, accompanied by reductions in the complexity of on-premise business infrastructure).

Businesses and vendors must review their existing business processes, technology roadmaps and staffing plans, evaluate the impact SaaS could have, and make the necessary adjustments to their strategies and risk management plans.

---

[1] Worldwide and US Software as a Service 2005–2009 Forecast and Analysis: Adoption for the Alternative Delivery Model Continues, IDC, March 2005

[2] Saugatuck Technology, "Research Alert Survey of over 250 Senior Business and IT Executives.", March 7, 2007

Why the Traditional Software Model is Changing

Spending on software has grown rapidly in the last 3 decades, from less than $1 billion in 1970 to $138 billion in 2000[3]. Along the way, technology has evolved from mainframes to mini-computers to client-server and finally in 1991, to the Internet. The software industry also evolved across new categories of business users and business functions (use cases), but remained primarily focused on the enterprise (applications based on client server and LAN technologies). Consumers were left behind since they could not be connected on a single network.

The internet has changed this forever, by connecting business users and consumers on the same network, allowing them to communicate, collaborate, transact and access information and services. During the latter half of the 1990's the resulting exponential increase in user types drove both - the invention of new use cases (e.g. customer self-service) and improvements in old use cases (e.g. order management), leading to hyper growth.

The Crises Driving Industrialization

Now, a decade later, the industry is maturing after emerging from the post-boom recession in 2001. Rationalization and consolidation has set in, driving the industrialization of value chains in the software industry, a classic pattern for the tail end of a wave. Driving factors include a reduced level of confidence among enterprise senior leadership as regards IT spending, due to the following crises. [4]

- **The "good enough" crisis**

  Every innovation cycle culminates in the "good enough" crisis in which product differentiation becomes a challenge, causing customers to lose interest in upgrades to existing software assets.

- **The "IT Does Not Matter" crisis** [5]

  Nicholas Carr's book has caused enterprise senior leadership to re-evaluate their perception of IT investments. Per Carr, the strategic importance of IT is not growing but shrinking as it becomes more standardized and affordable such that it is no longer a competitive differentiator. This view has resulted in a general disillusionment with IT.

- **The complexity crisis**

  A term coined by IDC Research, this refers to a desire by enterprise IT management to strive for simplicity in enterprise software. Reducing complexity produces immediate savings by reducing the need for knowledge workers and reducing mistakes, thus saving time and money.[6]

As a result, businesses are resisting high priced software with its high post-sales costs for deployment, operations and maintenance. Software pricing and sales have since come under increasing pressure as customers look for simpler, cheaper alternatives, paving the way for a new disruptive technology.

---

[3] Martin Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry* (Cambridge: MIT Press, 2003), 14-15

[4] "*The Future of Enterprise Software – How software companies can achieve high performance in an era of disruptive change and uncertainty*", Accenture, 2005.

[5] Nicholas Carr , *Does IT Matter*, HBS Press, 2004

[6] "Five Segments Will Lead Software Out of The Complexity Crisis", Anthony C. Picardi, IDC, Dec 2002

Rebuilding the Value Chain

Since 2001, software vendors have been challenged to get traction around upgrade cycles for new products, since previous versions already exceeded the customer's requirements (the "good enough" crisis). In this mature market, the emphasis has shifted from proprietary product innovation to rebuilding the entire value chain to achieve greater efficiencies by leveraging the internet. Maximizing speed to value and minimizing the customer's Total Cost of Ownership (TCO) have become priorities.

- **Optimizing 'speed to value' (vendor perspective)**

  Pre-sales activities (the software development process) have traditionally been tightly coupled and managed through in-house development teams. The new process provides for a loosely coupled structure, facilitated by SOA architectures and open standards, facilitating reuse and allowing software vendors to distribute work and achieve cost efficiencies by utilizing low priced resources anywhere in the world.

  A flat world, facilitated by the internet, makes it easier to leverage geographically distributed, low-priced resources. Software companies have already learnt how to organize this labor into virtual web services factories, forming a modernized supply chain capable of building low priced, high quality software modules which can be centrally integrated into a final marketable product.

- **Minimizing TCO by targeting reduction of post-sales costs**

  Post-sales activities (software distribution, systems integration and deployment, customer operations, service, maintenance and upgrade) in the traditional software model were loosely coupled, labor intensive and inefficient. 50-90% of a customer's TCO arose from post-sales services provided by consulting firms, channel partners, and internal IT operations.

  The new trend is toward re-coupling these downstream activities to gain better efficiencies, and an improved customer experience. In the extreme case, hosting vendors manage all these activities in-house, controlling distribution, access, authorizations, and monitoring application usage by customers, while collecting data which helps them further improve their products. These upgrades are then deployed transparently to the customer, replacing the old batch upgrade cycles with a much smoother process. This also removes the need for the customer to invest in additional infrastructure.

Achieving these goals requires changes not only to the underlying technology and standards, but also to the existing business models of all players in the software ecosystem. The Internet facilitates these changes by promoting global competition and collaboration.

SaaS meets a market need

The term *Software as a Service* (SaaS) started to circulate in 2000/2001[7] and was presented conceptually as a low-cost way for businesses to obtain the same benefits of commercially licensed, internally operated software, but without the associated complexity and high initial cost. Since then, it has taken on wider implications and has become the industry preferred term, applicable to business and consumer applications, generally replacing the earlier terms Application Service Provider (ASP), On-Demand, Online Services and Utility Computing.

[7] SIIA (Software & Information Industry Association) published white papers on SaaS in 2001.

SaaS is a vendor-hosted model for delivery of software applications over the internet, usually (but not always) to a browser interface. It has caught the attention of businesses due to its many advantages which include speed to value, low Total Cost of Ownership (TCO), licensing flexibility, scalability of operations, and customer empowerment to hold the vendor accountable for performance.

A new customer has the option to 'try before you buy', incurs low startup costs as regards infrastructure and software licenses, by paying for access to the software over the internet, based on a recurring payments model (monthly or annual contracts). The vendor is responsible for provisioning new customers and end-users over the internet, scalability of infrastructure to meet performance requirements, user training, software bug fixes and patches, functional upgrades and performance monitoring. The customer's main responsibilities include data migration to/from the SaaS application and integration with in-house systems based on a vendor provided API.

Many types of software are well suited to the SaaS model, where customers may have little interest or capability in software deployment and operations, but do have substantial computing needs. Early application areas such as customer relations management, video conferencing, human resources, accounting, email and security are now widening to embrace new areas such as procurement and compliance management.

The ASP Model compared to SaaS

Offering software services over the Internet is not a new idea. The ASP model was adopted by several vendors in the late 90's, but it never took root in the marketplace for a variety of reasons, including vendor credibility, software quality, cost and performance related issues.

The primary differences between the SaaS and ASP models are as follows:[8]

- ASP applications were traditional single tenant, client server applications with HTML front-ends added to facilitate remote access to the application. However, the economy of scale and performance possible with the multi-tenant, net-native applications developed for the SaaS model were generally not achieved.
- ASP applications were generally hosted by third parties who did not ordinarily have specific application expertise. SaaS vendors generally have specialized expertise in the applications they provide.
- ASP applications were not written as net native applications. As a result, their performance was poor and application updates were no better than self-managed traditional applications. By comparison, current net-native SaaS applications are updated regularly, many daily.
- The ASP model did not have the ability to scale since vendors merely deployed one application instance on a server for each customer, just as a customer would deploy internally.

To summarize, although ASP's took a step in the right direction, they did not go far enough to develop the winning value propositions of the SaaS model.

[8] "Software as a Service: A Comprehensive Look at the Total Cost of Ownership of Software Applications", SIIA White Paper, Sept 2006

Understanding & Managing SaaS

Definitions are Fuzzy

There is no comprehensive definition for SaaS today, just conceptual descriptions of the model's fundamental characteristics. Market forces and technological innovation continue to collaborate to stretch the conceptual descriptions proposed by various analyst firms and vendors.

We reviewed descriptions from analyst and vendor firms and have reproduced a couple below as references.

Key characteristics of SaaS (Gartner)

1. The application is owned, delivered, and managed remotely by one or more providers
2. The application is 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. The customer may be able to extend the data model by using configuration tools supplied by the provider, but without altering the source code.
3. The application is licensed on pay-per-use or subscription basis. A perpetual license purchase is not considered SaaS.

Key characteristics of SaaS (IDC) [9]

1. Network-based access to, and management of commercially available (i.e., not custom) software
2. Activities that are managed from central locations rather than at each customer's site, enabling customers to access applications remotely via the Web
3. Application delivery that typically is closer to a one-to-many model (single instance, multi-tenant architecture) than to a one-to-one model, including architecture, pricing, partnering, and management characteristics
4. Centralized feature updating, which obviates the need for downloadable patches and upgrades.

SaaS applications are generally priced on a per-user basis, sometimes with a relatively small minimum number of users, and often with additional fees for extra bandwidth and storage. SaaS revenue streams to the vendor are therefore lower initially than traditional software license fees, but are also recurring, and therefore viewed as more predictable, much like maintenance fees for licensed software.

[9] Traudt, Erin; Amy Konary, "2005 Software as a Service Taxonomy and Research Guide", June 2005

The Multi-Tenant Architecture

A well-designed SaaS application is scalable, multi-tenant-efficient, and configurable[10]. The architectural skills required to design such applications are considerable because of the need to support potentially hundreds of thousands of concurrent users. Therefore, the cost of building, testing and operating the application can be high.

Economies of Scale

The SaaS single instance, multi-tenant model (Figure 1A below) is the most frequently used and relies heavily on economy of scale to control capital expenditures and operating costs. Capital expenditures relate to the physical infrastructure within the datacenter, including hardware (servers, storage, racks, local area networks, etc.) and software (operating systems, database engines, virtualization software, security software etc.), required to meet the requisite service levels.

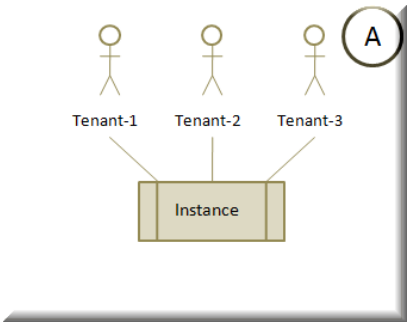

Figure 1A: Single Instance, Multi-tenant model

Operating expenditures relate to data center operating costs, including the managed services aspect of system monitoring, backups, disaster recovery planning and support, periodic server upgrades, application of operating systems and security patches, etc. to make sure that the infrastructure is being efficiently utilized and continues to meet customer needs.

Scalability

A disadvantage of the single instance, multi-tenant model is that the application can only be scaled by moving it to a more powerful server until diminishing returns make it impossible to add more power cost-effectively[11].

As the number of tenants and end-users continues to grow into the thousands, the vendor may be forced to run a load-balanced farm of identical instances (Figure 1B). This configuration is infinitely scalable. Another advantage is that the impact of a hardware or software failure can be localized and prevented from impacting all users on the system. Figure 1B shows a generic configuration with multiple application instances being managed by a tenant load balancer.

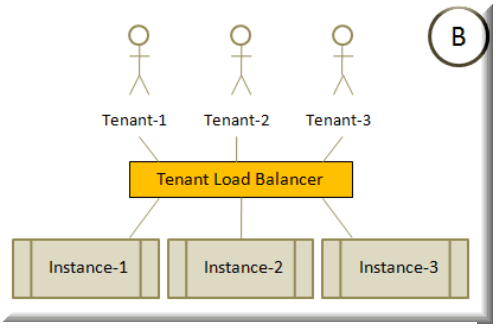

Figure 1B: Multi-instance, Multi-tenant model

10 "Architecture Strategies for Catching the Long Tail", Frederick Chong and Gianpaolo Carraro, Microsoft Corp., April 2006

11 The exception to this is when partitioning is used to manage database performance.

SaaS Application Configuration

Over the years, vendors have repeatedly cautioned customers against extensive internal customization of traditional software since it can lead to integration issues during a software upgrade (these typically occur every 3-4 years). With SaaS, since the vendor owns and manages the code for all tenants, the option to customize is no longer available to customers.

The advantage to customers is they no longer have to manage code changes, since it is the vendor's responsibility to patch, fix bugs and improve the functionality of the software. The advantage to vendors is they only have one code set to manage and this makes their task of building and deploying simultaneous upgrades to all customers much easier.

Customers do have access to a set of policy-based configuration options which allow them to tweak the software and its behavior to match their requirements. The table below shows a list of configurable items in a typical SaaS application.

Configurable Items	Benefits of Configuration
User interface and branding	Provides the customer with the ability to change things such as graphics, colors, fonts and embed logos on an as-needed basis.
Workflow and business rules	Allows the business to customize workflow based on its internal processes
Extensions to data model	Allows the user to add new fields and tables to the database, facilitating enhanced data capture, reporting and business intelligence capabilities (may need additional reporting and business intelligence software).
Access control	Allows each customer (tenant) to manage the roles and responsibilities of individual users and prevents unauthorized users from accessing sensitive data or making unauthorized changes to the system. This is a key requirement from the governance perspective (data protection).

Table 1: List of configurable items in a SaaS application

Although the software is standardized across customers, there are still opportunities for achieving competitive differentiation based on software configuration, quality of employees, user training, and integration with associated business processes.

SaaS Appliances

A SaaS appliance is a *single instance, single tenant* application and is a special category of SaaS which should appeal to enterprise customers concerned about trusting the SaaS vendor to host their data. The appliance is installed inside the customer's firewall or even a hosted virtual platform like Amazon's EC2. Vendors usually retain responsibility for managing patches and upgrades to the software via an Internet link.

The advantage of retaining the application inside the customer's firewall is that it overcomes the data security issue as well as the IT operations labor cost issue (the

vendor continues to remotely operate and manage the software).[12] Additionally, its performance is not impacted by network latency or surges in activity by other tenants. The customer may also request more control over the timing and frequency of upgrades, since the application is dedicated to a single customer.

Appliances come in the form of a physical box, with the onboard software integrated with its own operating system and possibly with its own database engine and middleware, usually all open source versions. Appliances can also be in the form of a complete software environment which is downloaded to a hypervisor (virtual platform) which allows multiple software systems to run on a single computer.

Appliances are simpler to implement because they come integrated with all the necessary software. They cost less because they do not require dedicated hardware platforms or infrastructure.[13]

Examples:

- Consider a device that includes a logistics application with a cached and periodically updated database. A shipping company might provide such a device to its large customers, so they can query the device for shipping information instead of hitting the shipping company's servers with thousands of individual queries a day, which could impact server performance.[14]

- The Google Search Appliance is a box physically installed inside the enterprise firewall. It is designed to crawl enterprise web servers, file servers, content management systems, relational databases and business applications (including employee directory and calendaring, CRM, ERP and business intelligence) and makes the information instantly available from a single search field in the browser. The appliance provides employees with real-time, secure access to information across the enterprise in more than 220 different file formats, and over 109 different languages.[15] Google also markets a Mini version for SMB's, which is priced more affordably than the enterprise version.

Several companies including HP, IBM, Oracle and Business Objects are working on appliances in the domain of business intelligence and data warehousing.

SaaS Integration Current Trends

SaaS integration appliances are a category of vendor solutions designed to make application integration easier through pre-built integration code based on vendor API's. These are plug and play devices designed from the ground up to remove the complexity and overhead associated with traditional integration solutions such as middleware and custom code development. They do not require installation or deployment of software and provide the capability to configure the appliance (for example, to process error handling scenarios).

---

[12] Phil Wainwright, "Can the Appliance put SaaS on-premise?", June 15, 2007, http://blogs.zdnet.com/SAAS/?p=345

[13] J. Bonasia, "Tech Firms Plug into Appliances", Investors Business Daily, July 19, 2007

[14] Software as a Service (SaaS): An Enterprise Perspective, Gianpaolo Carraro, Fred Chong, Microsoft Corp., October 2006

[15] Google website

SaaS integration appliances reduce complexity by automating four things - *connectivity, data transformation, workflow and management*, thus facilitating rapid integration of two or more applications, in a matter of days or weeks rather than the months it would otherwise take.[16] They serve 3 major functions:

- Data migration - getting information from a customer's backend systems into a SaaS system
- Synchronizing information between the SaaS system and the backend system, so that both systems contain current information for end-users
- Extracting information from a SaaS system – enables advanced reporting using business intelligence software and other applications.

Integration appliances are configurable (no coding is needed), minimize IT overhead (can be remotely managed), are paid for through subscription pricing, and are flexible, scalable and re-usable. However, they are not designed for environments which are heavily process centric (including business process management, multi-step processes which span several systems, human workflow, etc.). These areas are better served by middleware.

Integration appliances have experienced rapid adoption with customers looking for a faster and cheaper option to building custom code or using middleware to integrate SaaS applications such as SalesForce.com with backend packages (such as SAP, Lawson, JD Edwards, PeopleSoft, Oracle, etc.).

Future Trends

Industry analysts have already coined the term SaaS 2.0 to refer to the next generation of SaaS applications, which will presumably deliver significantly greater value via integration with other SaaS offerings as well as with legacy technologies, applications, data and business processes.

Today, the lack of a SaaS reference architecture, integration standards and formal methodologies for implementation and monitoring continue to be hurdles as regards SaaS application delivery, deployment, integration and performance management. Removing this hurdle will require a collaborative effort across the industry to build standards and arrive at repeatable, manageable and measurable processes. Middleware vendors will have an important role to play in developing products to facilitate the task of integrating SaaS products with in-house systems. The stages in this evolution may look as follows[17]:

Stage 1 – No formal architecture or methodology

Stage 2 – Ad-hoc approaches to architecture and methodology, some proprietary architectures

Stage 3 – Formal reference architecture and widely-adopted methodology for SaaS integration

Stage 4 – Established architecture and methodology with formal measurement programs

Saugatuck Technology estimates that until 2010, enterprises will lack the tools, methods and implementation resources to do SaaS integration effectively.

---

[16] Simon Peel, "Application Integration for a SaaS World", Cast Iron Systems, Jan 2007

[17] Saugatuck Technology, "SaaS Integration Platforms: The Looming SaaS Deployment and Integration Dilemma", Oct 6, 2006

Managing SaaS: The Changing Role of IT

Decisions regarding bringing a SaaS application in-house should always be made with IT involvement, so that technology, data, governance and roadmap issues can be discussed prior to finalizing decisions. Even though it is relatively easy to set up a new account with a SaaS vendor and terminate it if for any reason, both these events impact business data and IT labor, areas which are the responsibility of IT management.

IT's role in the consistent implementation of IT governance guidelines and stakeholder decisions makes them a key player in SaaS-related decisions. At stake is the strategic need to align corporate data management policies and standards, maintain business continuity and disaster recovery plans, and build technology roadmaps to support data migration, synchronization and business intelligence technologies and strategies. Operationally, there is a need to populate the SaaS application with existing business data, integrate the SaaS application(s) with internal systems, followed by measuring and managing vendor performance. IT will also need to plan for business data recovery if and when the agreement with the vendor terminates for any reason.

The complexity of managing the lifecycle of relationships with SaaS vendors in the context of an enterprise can be daunting. It is advisable to leverage an established capability model to ascertain that the organization has a structured means of evaluating the business need, the SaaS product, any supporting technologies, the process of building the SaaS vendor relationship and that it is able to plan, resource and manage all accompanying tasks.

Utilizing a Capability Model – The eSCM

IT Enabled Sourcing (or eSourcing) uses information technology as a key component of service delivery or as an enabler for delivering services, and is often provided remotely over data networks (such as the internet). Figure 2 shows examples of sourcing services. The scope of eSourcing includes the central 2 circles - *IT Sourcing* and *Task & Business Process Sourcing*.

The eSourcing Capability Model (eSCM) is a capability model created by the IT Services Qualification Center (ITSqc) at Carnegie Mellon University. SaaS belongs to the IT Sourcing domain (note the reference to ASP's in the IT Sourcing circle).

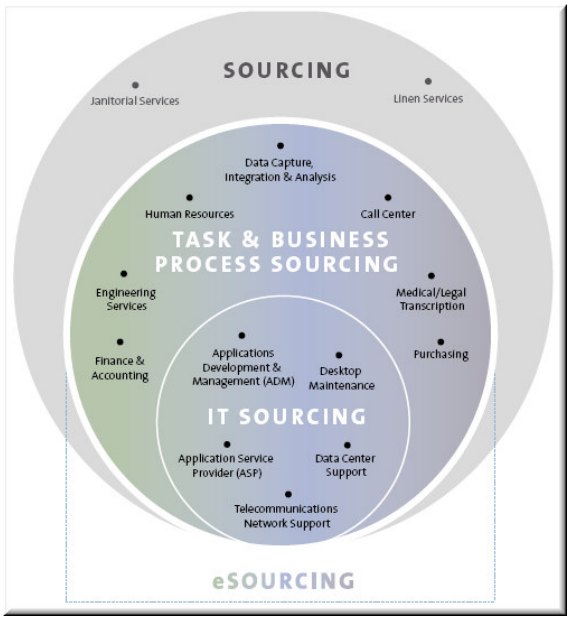

Figure 2: Types of sourcing services

The central 2 circles ('IT Sourcing' and 'Task & Business Process Sourcing') are covered by eSourcing. The outer circle is excluded from eSourcing.

Source: eSCM-CLv1.1, Part 1, p6

The model provides customers and vendors with an exhaustive coverage of the structure needed to communicate, collaborate and manage SaaS projects from cradle to grave. It is a preferred tool used by major consulting firms and outsourcing firms to guide eSourcing projects.

The eSCM documentation is downloadable free from the internet. It is composed of 95 Practices associated with successful sourcing relationships which are organized by:

- Sourcing lifecycle (5 phases)
- Capability areas (17 areas)
- Capability maturity levels (5 levels)

The model has 2 versions, one for client organizations and another for vendor organizations. The client version was designed to help improve client capabilities across the sourcing life-cycle and provide an objective means of evaluating sourcing capability. It addresses various tasks, from developing the organization's sourcing strategy, planning for sourcing and service provider selection, initiating agreements, managing service delivery and service completion.

Table 2 shows the Phases of the Sourcing Lifecycle, Capability Areas, and Capability Levels of the eSCM (client version), and the number of Practices associated with each.

		CAPABILITY LEVEL			
SOURCING LIFE-CYCLE	CAPABILITY AREA	2	3	4	TOTALS
50 Ongoing	Sourcing Strategy Management (str)	4	1		5
	Governance Management (gov)	2	5		7
	Relationship Management (rel)	2	3	2	7
	Value Management (val)		2	5	7
	Organizational Change Management (ocm)	2	4		6
	People Management (ppl)	2	2		4
	Knowledge Management (knw)	1	3	1	5
	Technology Management (tch)	3			3
	Threat Management (thr)	5	1		6
9 Analysis	Sourcing Opportunity Analysis (opa)	2	2		4
	Sourcing Approach (app)	4	1		5
20 Initiation	Sourcing Planning (pln)	5			5
	Service Provider Evaluation (spe)	3			3
	Sourcing Agreements (agr)	6	1		7
	Service Transfer (tfr)	4	1		5
11 Delivery	Sourced Services Management (mgt)	9	2		11
5 Completion	Sourcing Completion (cmp)	4	1		5
	Totals	58	29	8	95

Table 2: Overview of eSCM[18]

It is outside the scope of this paper to describe the model in greater detail. Eco-Bridge plans to release a more detailed paper on eSCM in August 2007.

[18] William Hefley & Ethel Loesche, "The eSourcing Capability Model for Client Organizations, (eSCM-CL) v1.1, Part 2, Practice Details", ITSqc

Market Adoption & SaaS Value Propositions

Market Adoption Trends

The SaaS paradigm has shown remarkable growth in traction over the last couple of years. Consider the statistics below:

- The worldwide SaaS market reached $6.3 billion in 2006 and is forecast to grow to $19.3 billion by year-end 2011, according to Gartner, Inc.
- Per IDC, 10% of the market for enterprise software will migrate to a pure SaaS model by 2009.[19]
- Credit Suisse's March 11, 2007 "On Demand Market Forecast" projects "On Demand software to grow at a CAGR of 36% to roughly $21 billion in 2011 from over $4 billion in 2006."
- A McKinsey survey in the fourth quarter of 2006 found that 61% of North American companies with sales over $1 billion plan to adopt one or more SaaS applications in 2007, a dramatic increase from the 38% planning to install SaaS apps in 2005.
- A Saugatuck survey indicated the number of companies over $1 billion in revenue that said they were planning to deploy SaaS for mission critical applications more than quadrupled over the previous year, from 13% to 53%.[20]

Adoption by Vertical

Figure 3 shows adoption by vertical based on a Gartner study. Technology, financial services and utilities are the leaders in SaaS adoption.

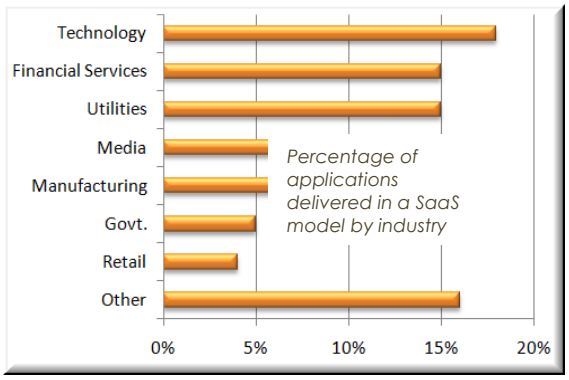

Figure 2: SaaS Adoption by Vertical Market

Data Source: Gartner

Adoption by Application Category

In general, SMB's are more likely than enterprises to implement a SaaS model, primarily due to the lower financial hurdles and the opportunity to leapfrog technologies at acceptable risk levels. Enterprises are generally more risk averse due to existing investments in core technologies and restrictive policies around data management.

---

[19] Worldwide and US Software as a Service 2005–2009 Forecast and Analysis: Adoption for the Alternative Delivery Model Continues, IDC, March 2005

[20] Saugatuck Technology, "Research Alert Survey of over 250 Senior Business and IT Executives.", March 7, 2007

Figure 4 shows major software categories (leftmost column) and high level indicators of the level of traction achieved by SaaS vendors in the enterprise and SMB markets:

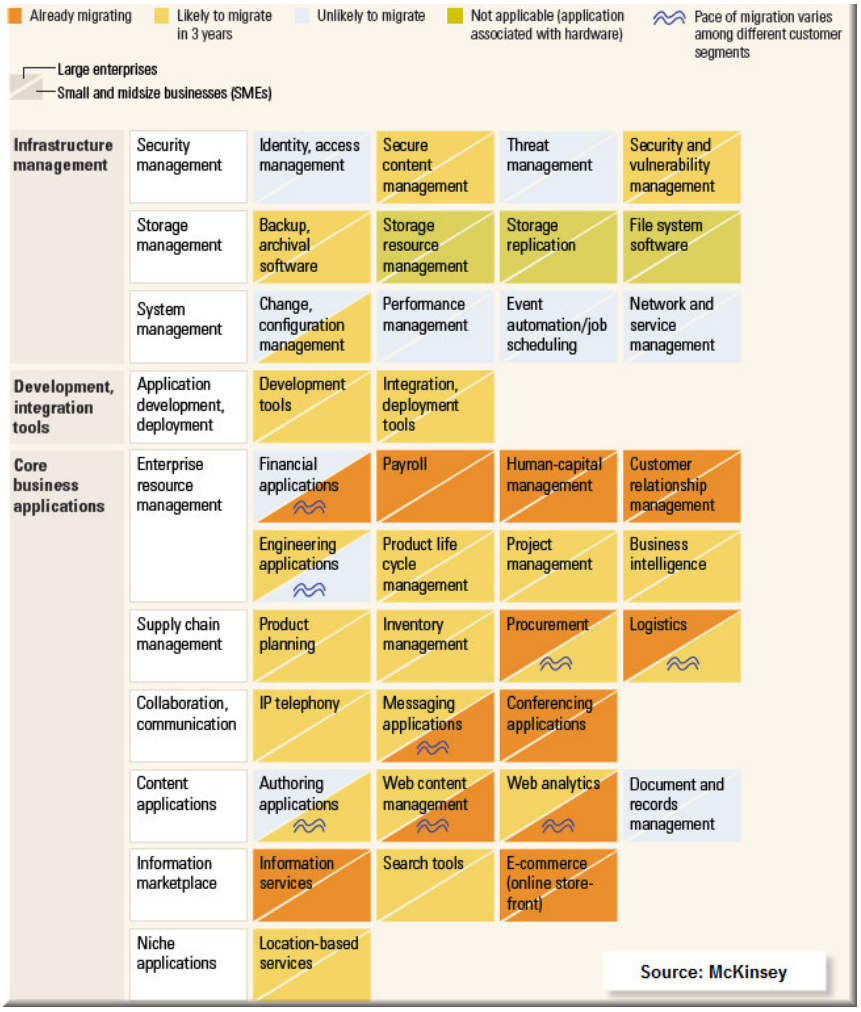

Figure 4: Migration of Applications from Traditional Delivery to SaaS[21]

SaaS applications are being used in the following domains in the Enterprise:

- ERP (payroll, HR, CRM),
- Supply chain management (procurement, logistics),
- Collaboration and communication (messaging, conferencing),
- Content applications (web content management, web analytics)
- Information-based applications (information services, eCommerce storefronts)

SMB's (unlike enterprises) have in the meantime, adopted SaaS not only for the above areas, but also in critical business areas such as financial applications, communications and marketing/sales applications.

[21] Abhijit Dubey and Dilip Wagle, "Delivering Software as a Service", The McKinsey Quarterly, May 2007.

SaaS Value Propositions

This section describes the qualitative and quantitative benefits of using SaaS products. Appendix A provides a summary table of the features and advantages of SaaS over traditional software.

1. SaaS overcomes resistance to new investments

In a previous section, we discussed the 3 crises contributing to market maturity and consolidation as regards the traditional software model:

- The "good enough" crisis
- The "IT Does Not Matter" crisis
- The complexity crisis

For certain types of software, customers no longer care that traditional software features and performance are better than corresponding SaaS offerings. Today's improved SaaS software is good enough to meet customer requirements and is rapidly becoming a substitute for traditional software, when its other advantages are taken into account (see sections below).

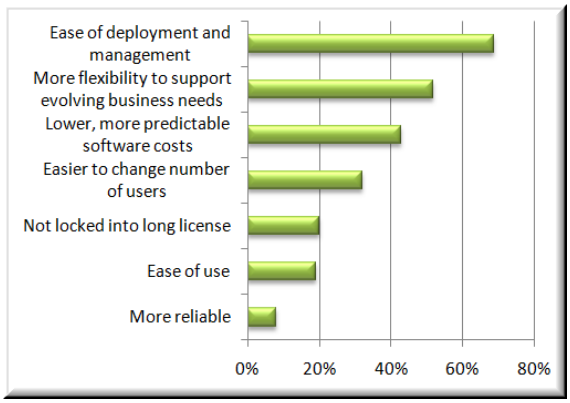

Figure 5: Survey Results: Why enterprises are adopting SaaS

***Data Source**: Information Week Research Software as a Service survey of 250 business technology professionals* [22]
Note: Multiple responses allowed
Base: 159 companies using or planning to use SaaS

2. Speed to Value

SaaS applications can typically be deployed in a matter of weeks to a couple of months, rather than the months or (sometimes) years it takes to deploy traditional applications. In most cases, the existing infrastructure at customer premises is sufficient to enable operation of SaaS applications. The customer does not need to build and execute strategic plans for infrastructure upgrades and license purchases based on assumptions of future growth, since the SaaS model enables them to pay only for current users and defer additional purchases until any additional users actually need access to the software.

A new customer can open an account over the internet, test-drive the software from the functional and configurability perspectives, subscribe by paying over the internet and start using the software within a few days to a few weeks. Pre-requisite activities prior to a launch must include a) migration of existing customer data into the SaaS application and b) integration with existing customer systems. These activities can be facilitated by SaaS integration appliances (described in an earlier section).

[22] Mary Hayes Weier, Lisa Smith, "Businesses Get Serious About Software As A Service" InformationWeek, April 14, 2007

3. TCO savings

The annual cost to own and manage software applications can be up to four times the cost of the initial purchase. As a result, companies end up spending more than 75% of their total IT budget just on maintaining and running existing systems and software infrastructure.[23] The key cost drivers include the cost of the software application, the hardware required to run it and the people services required to design, deploy, manage, maintain and support the application.

SaaS applications on the other hand, are billed on a subscription basis which includes all the above costs. In performing a total cost of ownership (TCO) comparison between traditional software and SaaS, the people services must be properly accounted for. Once this is done, the SaaS model typically wins the TCO comparison over the traditional software model.[24]

3.1 HIDDEN COSTS

Several hidden cost categories in a traditional software deployment (for example, CRM) cause it to be the more expensive option. These categories are:

1. *Needs analysis and site preparation* (strategy and business process consultation, site assessment & testing of existing equipment)
2. *Implementation/Deployment Costs* (dedicated hardware (servers, desktops, etc.), software licenses and gear for network upgrades, project costs including IT labor and consultant costs)
3. *Ongoing Operational Support* (basic datacenter operations, labor costs for internal IT staff and consulting resources, patches, upgrades, etc.)
4. *Strategic costs* contribute to the fully loaded TCO (downtime, advanced data center build out, real time dashboards, custom reporting, etc.)
5. *Upgrade costs* in the 3rd or 4th year of operation, can be significant

In the SaaS model, almost all startup costs are borne by the vendor except for (primarily labor) costs associated with data migration and application integration. The IT budget can therefore be stretched much further due to lower startup costs.

NOTE: There is a big difference between basic and advanced data center operations. Data center capabilities which provide 99.97% uptime are complex, very expensive and beyond what most customers can afford for an on-premises solution. It would cost millions of dollars in capital expenditures and personnel hours for a premises-based solution to match the software and hardware infrastructure of a hosted solution. Most customers cannot justify this cost and end up installing the minimal hardware and software required which may result in a lower uptime capability.

The cost incurred from downtime as well as the man-hours spent scrambling to restore the system can be sizable. The intangible costs incurred from lost productivity of end users and poor user adoption due to frustration of poor performance are hard to calculate[25].

A comparative view of the relative costs of traditional software versus SaaS is shown in Figures 6 & 7 (below).

---

[23] Timothy Chou, The End of Software, SAMS Publishing, 2005, page 6

[24] Adapted from "Software as a Service: A Comprehensive Look at the Total Cost of Ownership of Software Applications", Sept 2006, SIIA White Paper

[25] The Yankee Group, "Understanding Total Cost of Ownership of a Hosted vs Premises-based CRM Solution", 2004

3.2 DETAILED TCO BREAKDOWN – TRADITIONAL SOFTWARE MODEL

Figure 6 shows cost allocations in a traditional software deployment. Ongoing personnel costs are a significant portion of TCO (50-85%).

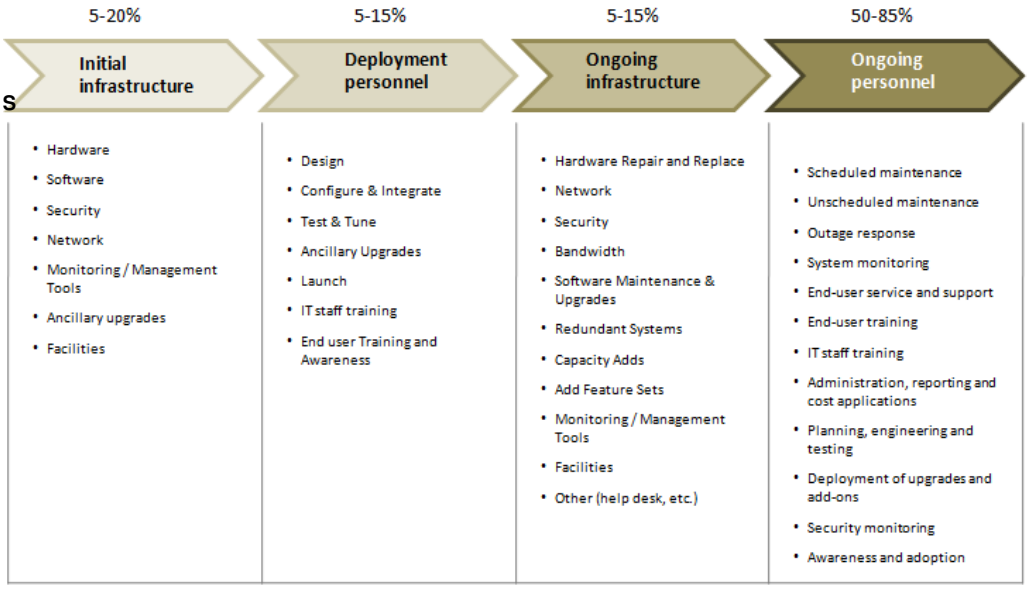

Figure 6: Summary of cost allocations of a traditional software deployment [26]

3.3 DETAILED TCO BREAKDOWN – SAAS MODEL

Figure 7 shows the corresponding percent costs for the SaaS model. Remember that the total IT budget required for a SaaS deployment is substantially lower than for a traditional software deployment (explained in a prior section). The biggest TCO factor is subscription fees (80-90%).

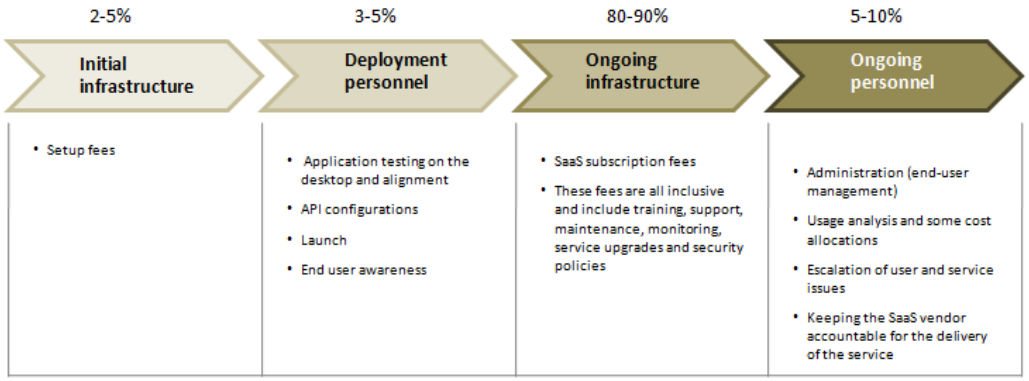

Figure 7: Summary of cost allocations for a typical SaaS deployment

[26] "Software as a Service – A Comprehensive Look at the Total Cost of Ownership of Software Applications", pp19, Sept 2006, SIIA.

4. Breakeven Point comparison

Another argument for 'owning' traditional software has been that even with the higher upfront costs, there is a breakeven point at which the traditional software becomes cheaper than the SaaS model.

IDC analyzed several SaaS versus traditional software installations over a period of 3 years and found that when hidden people costs and upgrade costs are taken into account, the breakeven point may never be realized.[27]

Note: Customers are cautioned to conduct their own analysis since pricing varies by application and vendor. Current discounted vendor pricing may impact this conclusion.

5. Vendor Accountability

The recurring subscription fee paid by SaaS customers allows them to hold a SaaS vendor accountable on a monthly basis, even if business conditions change and the customer has to add or shrink the number of seats. This is a boon to customers who normally have few options with a traditional software vendor after the sale is concluded. The terms of a typical SLA include a reparations policy which covers downtime, performance, and support. Triggering a documented condition in the SLA allows the customer to ask for a refund/penalty.

The customer also has the option to terminate service if dissatisfied for any reason and either switch SaaS vendors or bring the application back in house. Vendors will do their utmost to avoid such a loss of customer confidence since it impacts their recurring revenue stream and investor confidence, especially if other customers also decide to cancel their contracts. In such situations, financial institutions are likely to devalue the stock, making it much more expensive for the vendor to borrow money in the open markets.

6. Customers remain focused on core competencies

Most businesses simply want to use software for the benefits it provides, but they do not necessarily want to buy, customize and deploy it and then manage its ongoing operation, support and maintenance. If a business commits to traditional software, it requires the expense of purchasing the appropriate licenses, and the retention of an IT work force to provide all these services in-house. That takes away management focus and resources (financial and personnel) from core business activities and in many cases can make the business less nimble as its systems age and gradually face obsolescence, while making its business data less accessible.

For a business to become successful, its management team must focus on differentiating its core products and services from the competition. Anything outside this focus area is a distraction and many business authors[28] advise outsourcing non-core activities to one or more competent vendors.

Standardization of a business process based on a SaaS application is an efficient way for the business to quickly get that process up and running at minimum cost, without wasting time and resources on design, build and customization. Remember that even though the software itself is standardized, a business can still differentiate itself based on the quality of employees, user training, software configuration and efficient management of associated business processes.

---

[27] Robert Mahowald, "Do Service Providers Deliver Value and Reduce Enterprise Costs", IDC, 2003

[28] Geoffrey Moore, "Living on the Fault Line", Harper Business, 2000

7. Agility & Scalability

A business adopting the traditional software model may be forced to invest in infrastructure to support a large number of future potential users, even if not all of these users will use the application immediately. Additionally, businesses will often purchase user licenses for anticipated future users up front (to gain volume licensing discounts) even if it means that several licenses will sit idle, thus tying up budget resources without providing any benefit to the business.

With the SaaS model, the business only needs to purchase the number of licenses immediately required and defer purchase of the rest until later. It is up to the vendor to scale up the SaaS platform as demand grows to support new users at each customer (see section on vendor accountability).[29] This allows the business to be more agile in managing its budget and its licenses.

Another scenario in which the agility and scalability offered by the SaaS model are an advantage is when companies going through a merger/acquisition face significant issues related to the re-alignment of IT systems. Leveraging a SaaS application can bypass the need to consolidate or integrate various systems across the acquirer and the acquired companies.

8. The 'Long Tail'

The concept of 'The Long Tail'[30] highlights an economic model which states that when a product is priced affordably (even cheaply) and made accessible to all market segments, including geographically remote niche markets, the overall size of the resulting market can rival the size of the traditional market. Normally, the barriers to accessing such markets in the physical world are warehousing storage capacity and the limitations of physical distribution to customer locations. However if distribution is over the internet (as is the case with SaaS), these barriers are easily overcome.

There are thousands of small and mid-sized businesses (SMB's) who cannot afford the high prices of a traditional software deployment. For such businesses, SaaS offerings can be a boon by allowing them to initiate operations at short notice, with little investment and few long term commitments.

As mentioned earlier, a customer can subscribe for a trial, customize the software and convert to a full subscription by signing up over the internet, without the need for a sales visit from the vendor. Provisioning and billing are automated, as are software upgrades, data management, maintenance, etc. The SMB market therefore, represents a vast, untapped source of customers for SaaS vendors.

Aggregating the Long Tail is the proven basis of success for many existing internet companies such as Amazon, eBay, Real Music, etc. Distribution over the internet and pricing their products affordably has allowed them to tap markets which physical distributors have been unable to access.

---

[29] "Software as a Service – A Comprehensive Look at the Total Cost of Ownership of Software Applications", pp19, Sept 2006, SIIA

[30] Chris Andersen, "The Long Tail", Wired Magazine (Oct 2004).

Evolution of the Ecosystem

Investor Considerations

Valuation Trends - SaaS vendors

Valuations of SaaS vendors have been growing rapidly, as indicated by the stock prices of companies like SalesForce.com, Concur Technologies, Taleo Corp., etc. Technology investors like the recurring revenue model which provides more stability than the boom and bust cycles of traditional software.

Basis for Future Valuations of Software Vendors

About 50 percent to 80 percent of an enterprise software vendor's value is based on expectations of cash flows above and beyond current cash flows that the vendor will capture in the future (Figure 6). Hence, positioning for growth and addressing potential disruptive threats to growth, is likely to have a greater impact on a vendor's valuation than seeking to achieve higher profitability (via greater economies of scale or scope via consolidation).[31]

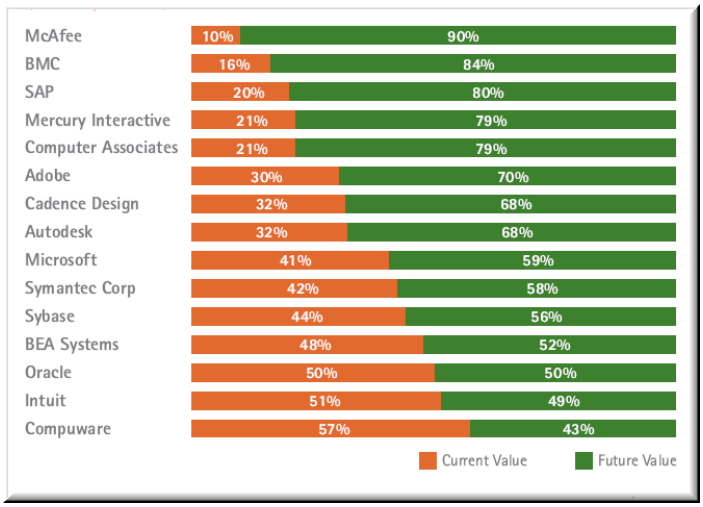

Figure 6: Current Value versus Future Value for Leading Software Companies

Future value is defined as Enterprise value minus the value of current operations and represents future incremental value the market expects the company to create, beyond the value delivered by current operations.

Positioning for future growth will require vendors to carefully evaluate their product roadmaps, growth prospects, and the impact of the SaaS paradigm.

Software Vendor Challenges

The Risks

SaaS is a disruptive and game changing technology from the customer's perspective, with its lower TCO, simplicity (of installation and operations), small footprint (browser), convenience (ubiquitous access through internet and wireless technologies) and ease of management (vendor SLA's which provide CIO's with 'one neck to choke'). Any lack of performance in comparison with traditional software is likely to be temporary.

[31] "*The Future of Enterprise Software – How software companies can achieve high performance in an era of disruptive change and uncertainty*", Accenture, 2005.

Incumbent software vendors are already experiencing challenges from SaaS startups. Having over-designed many of their products to the point where the majority of the functionality is unused (and often unknown) to a majority of customers, it is difficult to justify high prices and further upgrades (see the 3 crises described in Section 1), a factor which favors SaaS competitors from the economic perspective.

Migrating existing products to SaaS is likely to be an expensive exercise for vendors. Building a sustainable business will require major investment commitments (see *Options*, below) and rapid growth in the customer base to achieve the economies of scale necessary for economic viability. Increased sales and marketing expenses in the early stages are to be expected since SaaS is still fairly new and customers (SMB's and enterprise) will need to be educated by the sales force. These expenses are likely to drop in future, once market adoption increases to the point where new customers are willing to just sign up over the internet. Until then, profit margins will remain lower than at traditional vendors[32].

Given the expense, lower profit margins and the smaller market size currently associated with SaaS, incumbents have little motivation to consider transitioning existing, more profitable products. The downside is a loss of first mover advantage, since this leaves competitors free to build their customer base. This can lead to a reduced opportunity for future growth in the eyes of investors. Future money flow will be determined by customers and investors, not by vendors, regardless of size.

The Options

In order to transition over time to SaaS, incumbents will need to make some hard planning decisions about the cost and complexity of migration and the risk of revenue loss from cannibalizing existing products. They will need to balance this scenario against one in which they take no action, which will increase the likelihood of steady erosion of market share by startups, and reduced stock valuations by investors who see SaaS as the future direction of the industry. This is the classic Innovator's Dilemma[33].

Incumbents have a choice to wait and see if SaaS market traction continues and justifies the acquisition of a SaaS competitor, or start the transition process now. In the latter case, creating an entrepreneurial business unit which can stay focused on this emerging market will help mitigate some of the risks associated with remaining captive within an existing business unit which may be driven by a different set of customers, requirements and priorities. Migrating existing products to the SaaS model and hosting them is not likely to be cheap, although partnering (for e.g. with data center operators), can reduce deployment costs.

If incumbents choose to start the migration, existing products will need to be extensively re-architected as a part of the migration. Additionally, vendors will have to invest (or partner) to build data center capability, as well as build or buy OSS/BSS[34] systems to support SaaS deployments. Such systems are traditionally used in the telecommunications industry, which is itself facing commoditization of its landline business.

---

[32] Abhijit Dubey and Dilip Wagle, "Delivering Software as a Service", The McKinsey Quarterly, May 2007.

[33] Clayton M. Christensen, "The Innovator's Dilemma", Harper Business, 2000.

[34] OSS (Operational support services)—include account activation, provisioning, service assurance, usage, and metering.
BSS (Business support services)—include billing (including invoicing, rating, taxation, and collections) and customer management (which includes order entry, customer self services, customer care, trouble ticketing, and customer relationship management).

In a related market development, internet-based firms are already engaged in a race to increase data center capacity to support new applications and users since growth in the software industry in general is increasingly dependent on internet-based services of all kinds (including consumer services).

Table 2 shows data center investment trends for various companies.

Company	2006 Capital Expenditures (projected data, in millions)	Percent of Revenue
Amazon	$225.0	2.2%
eBay	$448.6	7.6%
Google	$1,424.9	13.9%
Microsoft	$2,642.0	5.8%
Yahoo	$622.1	9.1%

Table 2: Internet infrastructure build out (budgetary numbers) [35]

Channel Partner Challenges

The relationship between SaaS vendors and their channel partners will by necessity have to change in the future as channel partners evaluate their business model from the perspective of the new value-added services they can provide and changes to the compensation they can ask for.

Firstly, sales commissions per customer are likely to be smaller due to the recurring payment model and may be unsustainable for some channel partners. The channel partner must therefore adjust its revenue model accordingly, due to reduced profits per sale, and negotiate the duration of these commissions. Additionally, the channel partner must focus on rapidly growing its base of SaaS customers to make up any annual shortfalls in revenue.

Secondly, larger channel partners will most likely want to brand, price and host their own value-added packages, possibly by integrating SaaS offerings across multiple software vendors, as well as offering other value added services such as user training and support, data management, business continuity, disaster recovery, etc. This will necessitate either owning or renting data center space to host packaged SaaS solutions, as well as proprietary OSS/BSS systems which are connected to the OSS/BSS systems of the respective vendors.

[35] Goldman Sachs, Business 2.0 Analysis, June 2006 issue, pp20.

Conclusions & Recommendations

Conclusions

- As the software industry matures and consolidates, SaaS applications are well on their way to becoming a practical consideration for CIO's at small companies and enterprises. Risk averse customers tend to test out the application and the vendor by leveraging the former in smaller, relatively isolated functions which do not require integration with enterprise systems.
- Adoption of SaaS is higher at SMB's when compared to enterprises, since SaaS provides them with the opportunity to leapfrog technologies at low cost and with acceptable risk. Enterprises, with an installed base of traditional software which form their core technology infrastructure, are more reluctant to move quickly in the direction of SaaS adoption.
- Market revenue statistics and projections are healthy, but SaaS is expected to take at least a few years to mature as regards technology (reference architecture, integration standards, middleware), vendor offerings and the vendor ecosystem (channel relationships, SLA's, etc.).
- Incumbent vendors are still evaluating the impact of SaaS on their current products and investments, and looking for ways to hedge the future risk of large-scale market adoption of SaaS. Startups on the other hand, are in the process of rapid build out of SaaS products, using globally distributed software development teams and in some cases, virtual infrastructure capabilities (from providers such as Amazon and Sun Microsystems).
- The 'Long Tail' effect makes SaaS even more likely to succeed as a business model, since these applications are available over the internet to a larger market including SMB's (who may not be able to afford traditional software).
- Widespread adoption of SaaS will most likely have an impact on the hardware industry and associated vendors. We expect the demand for data center hardware and related software to continue to trend upward, while on-premise enterprise infrastructure and consumer hardware will benefit from reductions in complexity and reduced urgency for upgrades.

Customer Recommendations

- Although the SaaS ecosystem is still evolving, significant benefits are available to customers who decide to use SaaS products, including speed to value, TCO, quality of service, agility and scalability and vendor accountability. Cost comparisons between the SaaS model and the traditional software model show that when IT labor costs (50-85% of TCO for the traditional software model) are properly accounted for, TCO for SaaS deployments result in savings which can range from 10% to 50%.
- Customers looking for best practices in managing relationships with SaaS vendors should consider utilizing a capability model such as the eSCM to facilitate the lifecycle of the relationship and minimize business risks.
- The data isolation issue in the SaaS model can be resolved by compensating vendors to either host separate dedicated instances (at

higher customer cost) or installing SaaS appliances inside the firewall.

- We expect to see over time, a transfer of complexity in IT systems from the customer's premises to vendors' data centers. As a result, customer IT staff will initially be freed up to focus on more strategic issues, but over time, the need for IT operations staff, systems integrators and consultants will be reduced while demand for staff at data centers will increase.

Vendor and Channel Partner Recommendations

- Given the market momentum and interest building up around SaaS, incumbent software vendors must consider building SaaS versions of existing products, so that they do not lose first mover status to competitors. Cannibalization of existing products, although painful, will preserve market leadership and investor support, and facilitate access to the 'long tail' – viz. customers not currently being served by traditional software products.

- Incumbents currently have an advantage in building SaaS versions of existing applications given their resources and understanding of customer needs. However, changing to a SaaS model raises a new set of challenges. It requires building a new set of capabilities (re-architected products, investments in data center capabilities, OSS/BSS systems, online user training and support, performance guarantees, measurements, etc.).

- Due to lower profit margins in the SaaS model, vendors will need to decisively increase sales to gain economies of scale and increase profits.

- Vendors and channel partners will need to negotiate changes to their relationship and commission structures. Channel partner cash flows will be significantly impacted when the high commission sale in the traditional software model is replaced with smaller periodic transactions in the SaaS model. Commissions will drop dramatically as a result and transition to a monthly basis instead of large one-time payments. Channel partners must strive to decisively increase sales to maintain annual commissions.

- Vendors must try to provide customers with a variety of integrated SaaS products so that customers are not forced into building additional supporting systems to further process or manage data being generated or stored by the SaaS vendor. Vendors must look for:
  - Vertical integration opportunities to provide services such as data warehousing and business intelligence (provide customers with a one stop shop).
  - Horizontal integration opportunities to provide a portfolio of integrated SaaS applications (for example, integrated finance, HR, accounting capabilities) which permit them to maintain centralized business objects (such as the customer object), without fragmentation of these objects across multiple vendor and customer systems.

- Vendors must engage in cross-industry efforts to create reference architectures, integration standards and middleware products to facilitate evolution of applications toward a more mature SaaS landscape in future.

Appendix – Comparison of SaaS with Traditional Software

The table below is based on Eco-Bridge research and compares the traditional software model to SaaS based on a variety of categories (first column).

Category	Traditional Software	How SaaS is Different
Product Design	• The application is designed based on a client-server architecture and built for a particular operating system and runtime vendor environment. The vendor therefore has to build and maintain one version for each OS supported.	• The application is designed from the ground up to be web native. It is not a traditional client server application with a HTML front end and is independent of the customer's operating system. Design is based on a single instance, multi-tenant architecture.
Budget and Costs	• The typical IT budget is spent in 3 main areas: ↳ Software licenses & maintenance fees, ↳ Hardware & infrastructure, ↳ Services (setup and operations)	• A significantly larger percentage of the IT budget is available for software since most of the hardware and services costs are transferred to the vendor. • The cost of acquisition of software is replaced with paying subscription costs to access the software. Long term savings range from 10-50%, based on the economies of scale the vendor gains with the single instance multi-tenant model.
Software Licensing Costs	• High license acquisition costs. Perpetual licenses required regardless of duration or level of usage. • Licenses must be purchased in bulk to gain volume discounts	• Try before you buy, pay on subscription basis or utility-based billing. • Additional seats can be purchased at the same unit costs
Speed to Value	• Extensive effort (time and cost) spent on RFP process, vendor selection, business process analysis, software customization and final deployment	• Deploying a new SaaS application usually takes between a few days to a few weeks.
Access to Application	• The application (server infrastructure and clients) is housed within the enterprise. Clients access the server over the enterprise LAN using a rich client in most cases.	• Delivery is usually over the public Internet but can also occur over secure private networks, when the customer is willing to pay for this additional cost. End-users access the application using a browser.
Customization	• Customization can get very expensive if the customer requires the software to match existing business processes.	• Vendors do not permit customization of the source code, but generally offer the ability to configure the application (configuration options include: 1. user interface & branding, 2. workflow and business rules, 3. extensions to data model, 4. access control). This policy-driven configuration is facilitated by the underlying SOA architecture.

Category	Traditional Software	How SaaS is Different
Integration with Existing Systems	• Customer is responsible for integration with enterprise systems using vendor API's.	• Same. Application integration vendors now offer products which expedite the task of integrating SaaS applications with traditional software systems such as SAP, Oracle.
Upgrades	• Rapid obsolescence cycles force software upgrades. • Hardware upgrades are often driven by software upgrades when existing memory, storage or CPU capabilities are insufficient to run new software.	• Software is upgraded and maintained behind the scenes by vendor at no additional charge and with minimal disruption to customer operations. • Customer hardware is likely to have longer lifecycles since the end user simply needs a browser to access the application. Demands on hardware do not change significantly with time.
Operations & Support	• Operational personnel costs for traditional software are significant. Certain types of software can incur personnel costs of 5-7 times the cost of the software licenses over a 3-year period.[36]	• The vendor is responsible for all infrastructure and IT operations to support the application. This includes maintenance and upgrade, data management, performance management, customer training, help desk and tech support.
Data Management, Trust & Privacy	• Customer assumes responsibility for all data management activities including security, periodic backups, disaster recovery.	• The vendor assumes responsibility for most data management activities including security, periodic backups and disaster recovery.
Accountability of Software Vendor	• Few obligations exist after the sale of software licenses	• Greater accountability due to recurring payment model & SLA. Customer can stop paying or get a refund if the software does not function as specified or if SLA conditions are not met.

Source: Eco-Bridge Research

[36] "Software as a Service – A Comprehensive Look at the Total Cost of Ownership of Software Applications", pp13, Sept 2006, SIIA

About Anand Divekar

Anand Divekar (MS, MBA) is Managing Principal at Eco-Bridge LLC., and writes and speaks on topics of interest to the telecom/wireless, internet and media/content industries. He specializes in the business management of technology. His focus is on providing clients with analysis, insights and management services in a range of areas including financial analysis, market research/analysis (technology & market trends) and information technology (strategy, business alignment, transformation & effectiveness).

Mr. Divekar has worked with Fortune 500 clients in a range of areas including analysis of technology trends and international market trends, M&A analysis, governance and risk management, outsourcing management, business process and technology alignment, utilization of best practices, etc. He has also managed PMO's and project portfolios for clients including business process & IT transformation, cost reduction, post-merger systems consolidation, national program launches, development of award-winning custom software, package deployments, etc.

Mr. Divekar is an alumnus of Bearing Point, Deloitte Consulting and Verizon Science & Technology. He holds a B.S. in Engineering from the Indian Institute of Technology, M.S. in Computer Science from Clemson University, S.C. and an M.B.A. from PACE University, NY.

Contact information:

Email: Anand.Divekar@eco-bridge.com

About Eco-Bridge

Eco-Bridge LLC. is a management consulting firm specializing in the business management of technology. Our expertise is in helping companies increase revenues, improve efficiencies and manage risks. We have helped companies like AT&T, Cingular, Qwest, T-Mobile, KKR, Dow Jones, Screen Actors Guild (SAGPH), Safeco, Areva T&D, GM, Delphi and others. How can we help you?

Business consulting services

- Market and industry research and analysis (international)
- Financial services (M&A)
- Expert witness services
- Governance and Risk Management services

Technology consulting services

- IT Strategy and Business Alignment
- IT Transformation
- IT Effectiveness and Cost Reduction
- Outsourcing Management
- Program Management