Asia Pacific Journal of Information Systems
Vol. 20, No. 2, June 2010

Does the SaaS Model Really Increase Customer Benefits?*

Seojung Yang**, Byungjoon Yoo***, Jungjoo Jahng****

Software as a service (SaaS) is one of the most-talked about trends in IT. Unlike traditional perpetual licensing model, software applications are sold on subscription bases and services are provided over web by the vendors. It is said that SaaS can make vendors to invest more on R&D than on marketing while offering its customers better quality software applications at lower costs.

By empirically comparing vendors providing their software applications either by SaaS or by traditional perpetual licensing model, we examine whether or not SaaS really increases overall customer benefits in terms of cost efficiency, software quality, and customization. We show that SaaS may not provide better quality or cost efficient software applications than perpetual licensing does. Then we provide two practical tools which are useful for customers to evaluate whether SaaS is better than perpetual licensing for the purposes of software applications they want to adopt.

Keywords : IS Management, Software-as-a-service, Software Quality, Cost Efficiency, Perpetual Licensing

* Byungjoon Yoo thanks the Management Research Center at Seoul National University for grant funding.
** Won Health System, Seoul, Korea
*** Corresponding author, Graduate School of Business, Seoul National University, Seoul, Korea,
**** Graduate School of Business, Seoul National University, Seoul, Korea

I. Introduction

Software as a service (SaaS), also known as "software on-demand", is one of the most-talked about trends in IT. SaaS is a new means of software licensing, which is different from a commonly used perpetual licensing model, called on-premise model. Traditionally, in a perpetual licensing model, a software vendor sells a software license which then the client implements and maintains in its own data center. On the contrary, in a SaaS model, the vendor (or the service provider) hosts the system on its own computers in its own data center and provides access to the system via telecommunication networks on a subscription basis. In short, SaaS turns software from a license sale to a subscription service.

The concept of SaaS goes back to the early days of business computing and more recently to the application service providers (ASPs) of the late 1990s. For this reason, many equate SaaS with the ASPs. However, there are three key differences between ASP and SaaS [SaaS Executive Council, 2006]. First of all, SaaS solutions are web-native applications. While applications that ASPs provide are often standard client server programs adapted for a hosted environment, SaaS applications are new versions developed specifically to be browser-based, making them more responsive, flexible, and scalable [Kaplan, 2007; Morris, 2007]. Secondly, a business model of SaaS is based on a utility style, pay-as-you-go approach. Whereas the ASP model requires an up-front license fee to the publisher and a subscription fee to the ASP, SaaS involves subscription payments to the publisher [Choudhary, 2007]. At last, SaaS is a multi-tenant architecture model, while the traditional software model is an isolated single-tenant model [SaaS Executive Council, 2006]. In a single tenant model, a customer buys a software application and installs it on a server which runs that application only for the single end-user group. However, in a multi-tenant model, the physically backend hardware infrastructure is shared among many different customers. Early successful adopters of the SaaS model include pure-play vendors such as Salesforce.com, NetSuite, and Webex. Salesforce.com offers on-demand CRM functionality built on its infrastructure and delivers to the users over the internet for a flat per-user, per- month fee. Its annual revenues grew from $50 million to $300 million (for year 2007) over the last five years. NetSuite offers on-demand enterprise resource planning (ERP) software. NetSuite reported revenue of $108.5 million for 2007, which is up 62 percent from sales of $67.2 million for 2006.

Prior academic literature on SaaS is quite limited and mainly focused on analyzing the SaaS model from a software vendor's point of view. Choudhary [Choudhary, 2007] compares perpetual software licensing to SaaS with a focus on the publisher's incentive to invest in product development. He concludes that the SaaS licensing model brings about greater investment in product development which leads to higher software quality in equilibrium as compared to perpetual licensing. There are also other articles focused on optimal pricing strategies under a pay-per-use model of software. However, most of them are not directly focused on the SaaS model. Rather, the articles are about early models of on-demand comput-

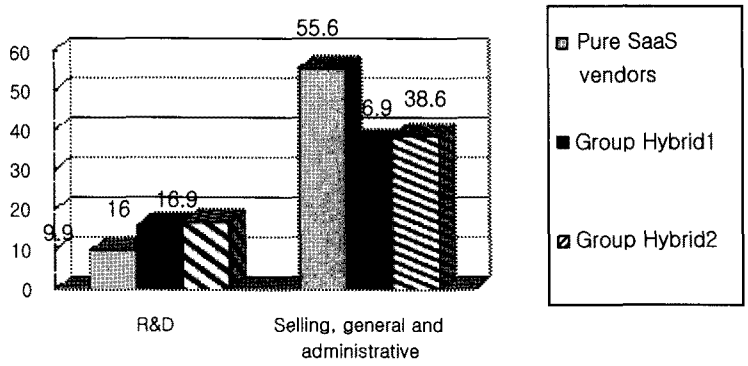
ing and ASP. Other articles from IT-related magazines are mostly focused on the cost efficiency of the SaaS model compared to the traditional on-premise model. In most of those articles it is said that SaaS is not only more profitable to software vendors but also more cost efficient to customers compared with a traditional licensing model because the SaaS model does not require large upfront costs and the maintenance fees are all included in the subscription fee. In their report, Gartner 2008 Top IT Trends, Gartner predicts that by 2012, at least one-third of spending on business application software will be in the form of service subscription rather than product license [Gartner, 2008]. While prior studies are mostly optimistic about SaaS' potential to increase customer benefits, here we examine whether or not SaaS really increases overall customer benefits in terms of cost efficiency, software quality, security, service availability, and customization.

Ⅱ. Investment in Software Quality

According to Choudhary [Choudhary, 2007], SaaS application vendors have higher incentive to invest in product development which results in higher software quality and increased social welfare. However, when we see the real figures, Choudhary's argument cannot be always supported. In this paper, we compare investments in product development of pure SaaS vendors and on-premise or hybrid model vendors who have on-premise and SaaS businesses together. We selected seven representative on-premise or hybrid SW vendors (Group Hybrid 1) from software top 100 software vendors reported by SDTimes in 2008 [Schell, 2008] and six pure-play SaaS vendors (Group Pure SaaS) from IDC and OliverWyman reports [Konary, 2004; Calhoun and Chaudhury, 2008]. For analysis, we grouped on-premise model vendors and hybrid vendors together because recently most traditional vendors have started offering both on-premise models and SaaS models. Some traditional vendors entered the SaaS market by developing their own new software while others such as IBM and Cisco did so by acquiring pure SaaS vendors. IBM acquired Corio in 2005 and Cisco acquired WebEx in 2007. Both Corio and WebEx are renowned SaaS vendors. In addition, we selected seven hybrid vendors (Group Hybrid 2) from software top 100 (Software Top100) to compare them with Group Hybrid 1. Companies in Group Hybrid 2 are those whose sizes are smaller than those in Group Hybrid 1 and similar to sizes of Group Pure SaaS. From the comparisons of these three groups we want to separate differences in investments resulting from firms' size and types of businesses. The company list of our analysis is presented in <Table 1>. We collected income statements of these companies and calculated the proportion of research and development (R&D) in percent of total revenue. In addition, we also calculated the proportion of selling, general and administrative expenses in percent of total revenue as they can give extra implication for the comparison of cost structures. We collected their income statements from 2006 to 2008. After analyzing R&D cost and selling, general and administrative expenses in percent of total revenue for each year, we calculated the total ave-

<Table 1> The Company List of Our Analysis, 2006~2008

Group SaaS (Pure)		Group Hybrid 1 (Big)		Group Hybrid 2 (Small)	
Lists	Percentage in revenue	Lists	Percentage in revenue	Lists	Percentage in revenue
Dealer Track	R&D: 4.8%	MS	R&D: 14.1%	Corel	R&D: 16.6%
	SGA: 42.9%		SGA: 29.9%		SGA: 42.6%
Kenexa	R&D: 8.4%	Oracle	R&D: 12.5%	Serena	R&D: 14.0%
	SGA: 43.1%		SGA: 25.3%		SGA: 35.9%
Taleo	R&D: 19.0%	SAP	R&D: 14.2%	WindRiver	R&D: 25.0%
	SGA: 50.6%		SGA: 26.2%		SGA: 49.7%
Omniture	R&D: 11.8%	Computer Associates	R&D: 13.6%	Ancys	R&D: 16.1%
	SGA: 59.6%		SGA: 47.5%		SGA: 30.3%
Salesforce	R&D: 8.9%	Adobe	R&D: 19.6%	ACI	R&D: 12.3%
	SGA: 66.0%		SGA: 40.8%		SGA: 42.8%
Vocus	R&D: 6.7%	Vmware	R&D: 22.0%	JDA	R&D: 15.9%
	SGA: 71.2%		SGA: 44.1 %		SGA: 28.8%
		Symantec	R&D: 16.1%	Autonomy	R&D: 18.1%
			SGA: 44.5%		SGA: 39.8%



<Figure 1> Comparison f Cost Ctructure for Pure SaaS, on-premise, and Hybrid Vendors

rage for the operating years of each company. Then, we calculated the average of each group. The average amount spent in R&D and selling, general and administrative (SGA) in percent of total revenue is presented in <Figure 1>.

The variation rate ranges from 19 to 4.8 for R&D and from 71.2 to 42.9 for SGA of Group SaaS (Pure). The variation rate ranges from 22 to 12.5 for R&D and from 47.5 to 25.3 for SGA of Group Hybrid 1 (Big). The variation rate ranges from 25 to 12.3 for R&D and from 49.7 to 28.8 for SGA of Group Hybrid 2 (Small). Comparing the cost structures of the two groups, we can see that the average percentage of

costs spent in R&D is six to seven percent higher in Group Hybrid 1 and Hybrid 2 than in pure-play SaaS vendors' group. On the other hand, the average percentage of costs spent in selling, general and administrative is about thirty percent higher in pure play vendors' group than in Group Hybrid 1 and Hybrid 2. Descriptive statistics are shown in <Table 2>.

<Table 3> shows the results of one-sample Kolmogorov-Smirnov Test. The test results indicate the variables (R&D and SGA) are normally distributed within each group and that each group has the same standard deviation for the two variables.

From observing descriptive statistics, we can see that SGA costs spent by Group Pure may be greater than those by Group Hybrid 1 while R&D costs spent by Group Pure may be less than those by Group Hybrid 1. These are the opposite of the results expected by Choudhary [2007]. We hypothesize as follows:

H1: There are no significant differences between the mean average percentage of costs spent in R& D of Group Pure and that of Group Hybrid 1

H2: There are no significant differences between the mean average percentage of costs spent in SGA of Group Pure and that of Group Hybrid 1

If H1 and H2 are rejected, we can say that SaaS companies may not provide greater services and benefits to customers when they spend more on SGA and less on R&D. However, in terms of generalizing these, there may be an alternative argument about our results. Most companies of Group Hybrid 1 have much higher total revenues compared to those of Group Pure. Thus, we expect arguments that

<Table 2> Descriptive Statistics, 2006~2008

	Group SaaS (Pure)		Group Hybrid 1 (Big)		Group Hybrid 2 (Small)	
	R&D	SGA	R&D	SGA	R&D	SGA
Mean	9.933%	55.567%	16.014%	36.9%	16.857%	38.557%
Maximum	19.0%	71.2%	22.0%	47.5%	25.0%	49.7%
Minimum	4.8%	42.9%	12.5%	25.3%	12.3%	28.8%
Std. Dev.	5.0182%	11.9168%	3.5078%	9.4448%	4.0484%	7.4218%
Observations	6	6	7	7	7	7

<Table 3> One-sample Kolmogorov-Smirnov (K-S) Test Results

	Group SaaS (Pure)		Group Hybrid 1 (Big)		Group Hybrid 2 (Small)	
	R&D	SGA	R&D	SGA	R&D	SGA
N	6	6	7	7	7	7
Kolmogorov-Smirnov Z	.608	.455	.712	.613	.634	.404
Asymp. Sig. (2-tailed)	.853	.986	.692	.847	.816	.997
Test conclusion	Not Reject	Not Reject	Not Reject	Not Reject	Not Reject	Not Reject

Note) ** p > .05/Not Reject: Test distribution is Normal.

the differences in mean average percentage of costs spent in R&D and SGA may come from the size effect of total revenues. So, we set Group Hybrid 2 which consists of hybrid SW vendors with smaller total revenues than Group Hybrid 1. The average of 2008 total revenues of Group Hybrid 2 was $331.7 million as in <Table 2>. The average of 2008 total revenues of Group Pure was $333.1 million. Then, we compare Group Hybrid 1 and Hybrid 2 to show that the differences in mean average percentage of costs spent in R&D and SGA do not come from the size effect of total revenues. We hypothesize as follows:

H3: There are no significant differences between the mean average percentage of costs spent in R&D of Group Hybrid 1 and that of Group Hybrid 2

H4. There are no significant differences between the mean average percentage of costs spent in SGA of Group Hybrid 1 and that of Group Hybrid 2

To test these four hypotheses, we will use ANOVA, Man-Whitney, and Kruskal-Wallis analysis to assess significance of differences in means across three groups. The ANOVA analysis can be used with variables of R&D and SGA normally distributed. And then, the non-parametric equivalent of one way ANOVA i.e., Man-Whitney and Kruskal-Wallis analysis will be checked again to make sure our results even though One-sample Kolmogorov-Smirnov test shows that the variables are normally distributed.

<Table 4> shows that H1and H2 are rejected ($p < 0.05$) while H3 and H4 are not rejected ($p > 0.05$) by all three analyses. These analyses of cost structures show that, regardless of sizes of companies, pure-play SaaS vendors spend a lot more proportion in Marketing and Sales (SGA) which brings benefits directly to vendors rather than customers. This result may be contrary to Choudhary's argument. Of course, we cannot generalize that all pure-play SaaS vendors invest less proportion of their expenses in R&D compared to traditional vendors. However, at least it can be said that his argument is not always the case.

<Table 4> ANOVA, Man-Whitney, and Kruskal-Wallis Test Results

	ANOVA		Man-Whitney		Kruskal-Wallis	
	R&D	SGA	R&D	SGA	R&D	SGA
N	6	6	7	7	7	7
Hypothesis 1 result (Asymptotic Significance)	Reject (.018)	Reject (.003)	Reject (.022)	Reject (.032)	Reject (.022)	Reject (.032)
Hypothesis 2 result (Asymptotic Significance)	Reject (.009)	Reject (.006)	Reject (.032)	Reject (.007)	Reject @(.370)	Reject @(.020)
Hypothesis 3 result (Asymptotic Significance)	Not reject (.711)	Not reject (.751)	Not reject (.701)	Not reject (.848)	Not reject (.701)	Not reject (.848)
Hypothesis 4 result (Asymptotic Significance)	Not reject (.711)	Not reject (.751)	Not reject (701)	Not reject (.848)	Not reject (701)	Not reject (.848)

** p > .05/@ Kruskal-Wallis statistic test for group 1, 2, and 2.

Ⅲ. Cost Efficiency for Customers

One of the most talked about benefits of the SaaS model is the cost efficiency of SaaS for customers. It is said that in SaaS model customers do not have to pay large upfront costs and maintenance costs thus resulting in cost benefits to customers. However, when we do the math, SaaS can be more expensive than people may think. Comparison of the cost per user for Siebel's on-premise software versus Salesforce.com's on-demand software is shown in <Figure 2> [Di Bona and Swanson, 2007]. Siebel offers CRM applications in traditional on-premise model whereas Salesforce.com offers only on-demand CRM applications. To calculate the license cost per user for Sieble, all of Siebel's license revenue since the company's inception were added and then the total were divided by 3.8 billion Siebel users deploying the software. The annual maintenance payment is calculated as 20% of that amount. Both of these assumptions are likely overly aggressive, since it is unlikely that all Siebel licenses purchased are still deployed today, and it is estimated that Siebel's true maintenance rate is closer to 17~18% of the initial license fee rather than the 20% used in the calculation. To calculate the annual cost of Salesforce.com's software, the base cost of $65 per user per month for Professional Edition was multiplied by 12 months to get an annual rate of $780 per user. However, Professional Edition is the low end of Salesforce.com's product suite and a user getting a solution functionally comparable to Siebel's offering likely pays more than $65 per month, an assumption again favoring Salesforce.com in this analysis.

To see the net present value of software cost, the costs of year 2-4 was discounted with a discount rate of 6%. We can see in <Figure 2> that NPV of the total cost for Siebel's license fee plus annual maintenance fee per user, $2,456, is less than NPV of the total cost of Salesforce.com's subscription fee, $2,865 [Di Bona and Swanson, 2007]. Therefore, the predominant conception of SaaS being cost efficient to customers cannot be supported in this case. Moreover, there are often hidden costs such as add-on costs and integrations costs for a SaaS offering. Sometimes, vendors charge to configure the software, implement the database, or offer trainings. In some cases, vendors charge an ad-

	Year				Total
Siebel	1	2	3	4	
License	1,416				1,416
Maintenance	283	267	252	238	1,040
Total	1,699	267	252	238	**2,456**
Salesforce.com					
Subscription	780	736	694	655	**2,865**

Note) Costs are discounted by 6.0% discount rate as Net Present Value.

<Figure 2> Software Cost Comparison: On-Premise (Siebel) vs. SaaS (Salesforce.com) ($ per user)

ditional $18 to $25 per user per month to stage and test the software. In addition, if a user wants to add support for handhelds and other mobile devices, those costs can escalate to $45 per user per month [Hoffman, 2006]. Moreover, integration cost, which is required to integrate a SaaS-based application with the pre-existing customer's environment, can be substantial [Gruman, 2007; Seidmann and Ma, 2004]. Integration issue will be discussed in the later section of this paper.

Ⅳ. Security, Service, Availability, and Customization

There are other important issues concerned with the SaaS model: security, service availability, and customization. The costs of those three are mostly intangible costs. Thus, it is difficult for a company to exactly calculate them. However, it is important to consider these factors when evaluating the SaaS model.

Security issues of SaaS have been one of the major hurdles in its mass adoption by the customers [Chaudhary, 2006; Donston, 2008; Kaplan, 2007]. The primary dogma that once prevailed in customer mindsets was the security concerns associated with the SaaS delivery method. The idea of putting mission critical data into an unknown server belonging to a third party was a 'no-no' to many organizations [Chaudhary, 2006]. Some advocates of SaaS argue that security and privacy concerns can be associated with any model and not just SaaS, because a purchased application can be just as vulnerable to security threats as an application remotely hosted by a service provider. As the SaaS market grows, the fear decreases, but security concerns continue to come up in annual SaaS surveys.

Service Availability is another issue of the SaaS model [Chaudhary, 2006; Donston, 2008]. Availability of SaaS applications depends on uninterrupted operation of broadband Internet connection. As a typical internet connection between a client and a server runs through the network of several operators, there are multiple points of failures most of which are out of control of both vendors and customers. Thus, the SaaS model has higher risks in service availability compared to the traditional models.

Above all, customization is quite limited in the SaaS model [Lashar, 2008]. In the traditional on-premise model, users are used to having the capability to slice and dice and essentially customize their own enterprise application to their heart's content. However, in the world of SaaS, almost all new applications are built on top of a common application platform, and customization can only be done to a certain extent. Although nowadays many SaaS models offer a vast array of options, the combination of options a customer can select is still limited to those available in the standard solution [Seidmann and Ma, 2004]. Thus, for a customer who wants a highly customized product, SaaS may not be the answer since SaaS is more about standardization.

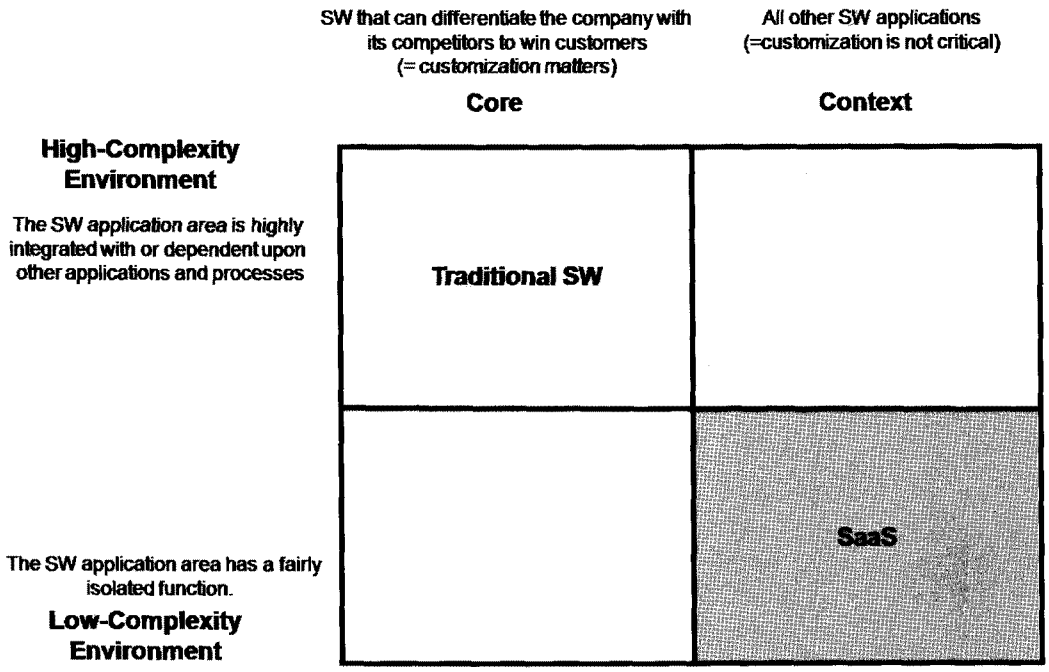
As we have seen in the previous sections, SaaS is not always beneficial to customers compared to traditional on-premise models. Thus, it is important for a company to fully evaluate whether or not SaaS is really appropriate for the company. There are many aspects that a company should consider before adopting this model. In this paper, we would like to suggest

two practical tools that any company can use to evaluate its fit with the SaaS model. Tools suggested here are very general, and they can be used with modification according to different industries or companies.

V. To SaaS or Not To SaaS: A Two-Dimensional Decision

We suggest a matrix that a company can apply when evaluating the fit of SaaS with its company (See <Figure 3>). Before adopting a new SaaS-based application, the management team can evaluate the application by two dimensions: 1) core/context analysis and 2) company environment related to the application area.

First, core/context analysis is based on Moore's core-versus-context theory which has great implications for the software market. Moore outlines the core-versus-context theory and argues that a company should mainly focus on its core activities to maintain competitive advantages [Moore, 2002]. Core activities are the services and products that can differentiate a firm's offerings in the eyes of the customer. For core activities, a company must allocate its best resources and differentiate as much as possible to affect customers' purchase decisions. On the contrary, all the other activities except for core activities are context activities. Context activities may be the key processes to maintain one's business but not competitive differentiators. For context activities, the goal is to manage effectively rather than to differentiate. Standardized process is one of the recommended means for context activities according to Moore. Moreover, a context task for one



<Figure 3> Core Versus Context-Complex Versus Simple Environment Matrix

firm can be a core task to another. Thus, outsourcing could be a great way to manage context activities. Except their core activities, most of companies' activities are context activities. So, software applications related to those may not serve as competitive differentiators to win customers over competitors. Therefore, it would be more cost efficient and effective for most companies and their context activities to manage their software related tasks through outsourcing. Cost related issues are discussed further in the following section.

Second, the other dimension a company should consider carefully is the integration of SaaS and the rest of the enterprise. As SaaS offering is designed to be a one-to-many solution, customers incur extra integration costs [SD Times, 2008]. When a company adopts a new SaaS-based application, it has to consider two integration problems: integration with their in-house traditional applications and integration among SaaS offerings [Chaudhary, 2006]. This puts an increasing burden on IT departments to use middleware and other approaches to solve integration problems. Tangible and intangible costs related to integration can be quite high. Therefore, the integration issue often hinders SaaS adoption, because it can be very consuming. It is often said that if the application area is highly integrated with or dependent upon other applications and processes, adopting SaaS can be inefficient compared to on-premise applications [Gruman, 2007]. In other words, SaaS is better suited to low-complexity environments where the number of touch points with other applications is low and the cost of integration is minimal. In high-complexity environments, it could be more expensive to integrate a SaaS solution into the larger IT ecosystem due to the limited ability to customize a SaaS offering [Di Bona and Swanson, 2007].

Therefore, before adopting a new SaaS-based application, the management team should evaluate the application considering the two dimensions. If the SaaS-based application lies on the grey colored area, it would be more effective to use a traditional on-premise application rather than a SaaS-based one. On the other hand, if a new SaaS-based application lies on the uncolored area, the SaaS model would be recommended. It may be difficult for a company to evaluate completely a SaaS application on the basis of the two dimensions indicated. It may need a third-party support to do the evaluation job. Conceptually, however, this matrix may give a clear insight into the decision of adopting the SaaS model. From Figure 3, it may seem that the on-premise application recommended areas colored grey is much larger than SaaS recommended area, uncolored. But this does not mean that, in the real market, SaaS has less potential than traditional model has. We should remember that most of software applications are related to context activities [SaaS Executive Council, 2006]. They do not provide competitive differentiation to win and retain customers. Moreover, the integration problem will become less critical as SaaS-based applications and related technologies develop. In fact, many SaaS vendors are very aware of this integration issue and putting much effort to solve this issue [Seidmann and Ma, 2004]. Thus, it is inappropriate to interpret this matrix as an indication of little potential usage of SaaS-based applications.

Ⅵ. Total Cost of Ownership

There are mainly three key cost drivers for software implementation: Product costs, hardware and infrastructure costs, and personnel costs. Product costs are the costs directly related to purchasing the actual application that a company uses for computing and information processing. Hardware and Infrastructure costs are the costs spent on servers, computers, networking components and other devices that provide users with access to the software. Personnel costs are the costs spent on internal IT staffs, vendors, and professional consultants to ensure the continuous availability of the application. In fact, it is said that the largest cost drivers for software implementation are the personnel costs. According to Gartner Inc. "more than 75% of the IT budget is spent just maintaining and running existing systems and software infrastructure" [SaaS Executive Council, 2006]. In traditional on-premise software models, vendors only offered the product itself, and it was the customers who were responsible for maintaining and managing software in their own servers. Traditional vendors sometimes offer professional support for the software but only with additional fees. However, in the SaaS model, software vendors not only initially provide the software product but also manage it in their own servers. In addition, they provide support services without additional fees. Instead, every cost related to software implementation is already included in the monthly subscription fees. Thus, when a company is to compare SaaS and on-premise applications, it is important to consider Total Cost of Ownership (TCO). The TCO assessment can ideally offer a final statement reflecting not only the cost of purchase but all aspects in the further use and maintenance of an application. Companies can easily make mistakes when comparing the costs of a SaaS-based application and a traditional on-premise application. While a SaaS subscription fee mostly includes product costs, hardware and infrastructure costs, and personnel costs, a traditional on-premise model usually include product costs only. Hardware and infrastructure costs and especially personnel costs related to traditional on-premise applications are not clear and difficult to evaluate. In the on-premise model, evaluating costs spent on internal IT staffs can be challenging as it is hard to say how much portion of total costs spent on IT staffs are related to software implementation. If a company mistakenly compares a SaaS subscription fee and an on-premise application purchase fee, the result would be misleading. The TCO comparison matrix provided below incorporates the key cost drivers mentioned above (See <Figure 4>). In addition, intangible cost category is included in the matrix because as mentioned earlier in this paper, there are other critical issues related to SaaS, which are not monetary concerns. These intangible cost variables may be challenging to clearly calculate in monetary terms, but we believe that it is still very important to consider these costs when calculating TCO.

Ⅶ. Conclusion

We have examined the effect of SaaS on customer benefits in the previous sections. First, we examined the investment in software quality.

	Initial Cost		Ongoing Cost					
	Setup & Deployment		Year 1		Year 2		Year n	
	On-Premise	SaaS	On-Premise	SaaS	On-Premise	SaaS	On-Premise	SaaS
Production Costs								
SW or License Cost								
Upgrades								
Platform SW								
HW & Infrastructure Cost								
Hardeware								
Storage (Datacenter)								
Other Facilities								
Personnel Costs								
Design and Engineering								
IT and Helpdesk Staffing								
End User Training								
Scheduled Maintenance								
Outage Recovery								
Purchasing and General Admin.								
Intangible Costs								
Customization								
Availability								
Interoperability								
Security								
Performance								
Capacity								
Opportunity Cost								
Initial and annual costs								
Total Cost after Year n								

<Figure 4> TCO Comparison Matrix

According to Choudhary [Choudhary, 2007], SaaS application vendors have higher incentive to invest in product development which results in higher software quality and increased social welfare. To empirically test his argument, we compared investments in product development from pure-play SaaS vendors and on-premise/ hybrid model vendors. Contrary to Choudhary's argument, a pure SaaS vendor spent less in R&D and more in marketing and sales than on-premise software vendors do. Secondly, to evaluate the cost efficiency of SaaS, we compared the cost per user for Siebel's on-premise software versus Salesforce.com's on-demand software. Contrary to the common conception that SaaS is more cost efficient than traditional models, after analysis, Siebel's cost per user turned out to be smaller than Salesforce.com's cost per user. We should consider other issues related to intangible costs such as security, service availability, and customization. Compared to the on-premise model, the SaaS model has weaknesses in those areas. As SaaS technologies develop, security and service availability issues may decrease. Some advocates of SaaS argue that with SaaS solutions, it is easier to protect against and resolve intrusions or internal bugs than with traditional solutions as the single code base that underlies the SaaS multi-tenant architecture makes patching much easier [Kaplan, 2007]. Moreover, according to SaaS providers, SaaS delivers the same service

availability with uptimes of 99.999 percent [Gruman, 2007]. However, security and service availability issues are still debatable. Also, there are customization issues. Since SaaS offers limited customization options, companies may instead be required to change their internal operations to fit the application. The traditional on-premise model has competitive advantages when it comes to customization.

Some analysts say that SaaS is disruptive enough to shake up the traditional software business [Wainewright, 2007]. According to Gartner, some parts of new business software, about 25 percent, will be delivered as SaaS [Gartner, 2008]. We expect that SaaS solutions will become more main stream, and therefore more enterprises will adopt them. But this does not mean that SaaS will bring disruptive innovation. Rather, as in <Figure 3>, SaaS and traditional on-premise models will co-exist as they serve different needs of various customers.

〈References〉

[1] Calhoun, J. and Chaudhury, R., "Software as a Service: The promise and the pitfalls for traditional software vendors," Oliver Wyman, *White Paper*, 2008.

[2] Chaudhary, R., "SaaS Security Concerns Are Cynical, Say Experts," *cxotoday*, available at: www.cxotoday.com/cxo/jsp/article.jsp?article_id = 73540&cat_id = 908, May 2006.

[3] Choudhary, V., "Comparison of software quality under perpetual licensing and Software as a Service," *Journal of Management Information Systems*, Vol. 24, No. 2, Fall 2007, pp. 141-165.

[4] Di Bona II, C. and Swanson, J., "Enterprise Software: Software as a Service-The End of Software or its complement?," *Black Book-The Long View: U.S. Perspectives*, June 2007, pp. 197-202.

[5] Donston, D., "Cleaning up with SaaS," *eWeek*, May 2008.

[6] Gartner 2008 Press Release, "Gartner highlights key predictions for IT organisations and users in 2008 and Beyond," available at: http://www.gartner.com/it/page.jsp?id = 593207, Jan. 2008.

[7] Gruman, G., "Get smart about SaaS," *www.cio.com*, June 2007.

[8] Hoffman, T., "Top 10 SaaS Traps," *Computerworld*, June 2006, p. 46.

[9] Kaplan, J., "SaaS: Friend or Foe?," *Business Communications Review*, June 2007, pp. 48-53.

[10] Konary, A.M., "Worldwide Software as a Service 2003 vendor shares: SaaS and enterprise ASP competitive analysis," *IDC #32055*, Vol. 1, Oct. 2004.

[11] Lashar, J., "The Hidden Cost of SaaS," *Customer Relationship Management*, May 2008.

[12] Moore, G.A., "Living on the Fault Line," *HarperBusiness*, 2002.

[13] Morris, J., "Sizing up SaaS," *Multichannel Merchant*, Feb. 2007, pp. 43-44.

[14] Seidmann, A. and Ma, D., "ASPs versus enterprise software solutions," *Workshop on Information systems and Economics, University of Maryland, College Park*, 2004, pp. 11-12.

[15] Schell, E., "Is SaaS for you?," *Multichannel*

merchant, April 2008, pp. 46-47.

[16] SD Times News Team, "SD Times 100: 2008," *SD Times*, available at: www.sdtimes.com/content/article.aspx?ArticleID = 32189, 2008.

[17] Software-as-a-Service Executive Council, "Software-as-a-Service; A comprehensive look at the total cost of ownership of software applications," *White Paper*, Sep. 2006.

[18] Software Top100, available at: http://www.softwaretop100.org/index.php.

[19] Wainewright, P., "How SaaS changes the SI universe," *ZDNet*, available at: http://blogs.zdnet.com/SAAS/?p = 344, 2007.

◆ About the Authors ◆

Seojung Yang

Seojung Yang received the B.S. degree in business administration and Master degree in Management Information Systems from Graduate School of Business at Seoul National University. She is currently working for Won Health System. Her research interests are on Software-as-a-Service, utility computing, virtual organization, and electronic commerce.

Byungjoon Yoo

Byungjoon Yoo is working at Graduate School of Business at Seoul National University as an associate professor. Before he joined Seoul National University, he worked at Korea University and Hong Kong University of Science and Technology. His research interests are on B2B e-commerce, online auctions and pricing strategies of digital goods such as software products and online games. He has published on these topics in journals such as Management Science, Journal of Management Information Systems, and Journal of Organizational Computing and Electronic Commerce.

Jungjoo Jahng

Jungjoo (JJ) Jahng is an associate professor of information systems in the College of Business Administration, Seoul National University (SNU), Seoul, Korea. Prior to joining SNU, he was a faculty member of Rensselaer Polytechnic Institute, U.S.A. He received the B.S. degree in business administration and Master of Business Administration (MBA) from Seoul National University, Seoul, Korea, and the Ph.D. degree in management information systems from the University of Wisconsin - Milwaukee, U.S.A. His research interests are in the domains of electronic commerce, IS strategy and IT-based innovation. His research has appeared in a number of journals such as IEEE Transactions on Systems, Man, and Cybernetics, European Journal of Information Systems, Journal of Information Technology, E-Service Journal and in refereed conference proceedings such as International Conference on Information Systems (ICIS). Dr. Jahng won the first Stafford Beer Award (Best Paper Award) in 2004 for his 2002 paper that appeared in the European Journal of Information Systems.