

Software Resource Management Considering the Interrelation between Explicit Cost, Energy Consumption, and Implicit Cost

A Decision Support Model for IT Managers

Jörn Altmann, Juthasit Robitratana

*TEMEP, College of Engineering, Seoul National University
599 Gwanak-Ro, Gwanak-Gu, Seoul 151-742, South-Korea*

1 Introduction

Although Software-as-a-Service (SaaS) adoption is expanding, the question whether SaaS is the right choice for an organization is still open. On the one hand, according to a 2008 survey by Cutter Consortium and THINKstrategies (2008), the top reason for adopting SaaS is lower IT infrastructure cost. On the other hand, depending on the maturity of the IT infrastructure, Kaplan (2007) pointed out that SaaS requires significant investments into the existing IT infrastructure. As proposed by SIIA (2006), the decision maker should use cost analysis in order to compare the total cost of both models. Choudhary (2007) suggested including the software quality of both licensing models into the comparison. In general, this shows that organizations need to consider a number of criteria related to their IT situation, in order to make optimal decisions about the type of software licensing model to be used.

Recently, this decision making process got even more complex by following a “Green IT” policy. Green IT refers to activities concerning sustainable IT resource management from an ecological perspective. IT managers have to optimize the use of IT resources such that fewer resources are wasted, in particular less energy. Since the execution of software determines the demand for computing resources, the software procurement process has to be included in this analysis as well. In turn, the software procurement process has to consider different software licensing models as alternative solutions. If the software is used rarely, the SaaS licensing model allows sharing of computing resources at times of low utilization. By aggregating the reduced demand for servers, it reduces the energy consumption caused by idle computing cycles. However, if the software needs the server continuously, the perpetual licensing model is more efficient. This situation corresponds

to the computing resource model analyzed by Risch et al. (2008). They discussed that Grid, as an on-demand model, is less expensive if the frequency of resource use is low. Based on this, we can state that reducing energy consumption is equivalent to finding the best software licensing model. The economic and Green IT motivations are aligned. In this paper, we show to what extent this is valid.

We use the Analytic Hierarchy Analysis (AHP) approach of Saaty (1980) to compare both software licensing models. The application of AHP for supplier selection has been illustrated in many fields (Wang et al., 2004; Ghodsypour and O'Brien, 1998; Lai et al., 1999). The advantage of AHP is that it can cover tangible and intangible criteria.

The main objective of this work is to propose the Software Licensing Selection Support (SL2S) model for selecting a SaaS Licensing (SaaS) model or a Perpetual Software Licensing (PSL) model. The SL2S model considers different types of criteria (i.e. explicit and implicit cost-related criteria and energy consumption) under Green IT policies and sustainable IT resource conditions (Jadhav and Sonar, 2009; Xin and Levina, 2008). Based on the criteria that influence the selection of the software licensing type, we perform a sensitivity analysis, to understand the dependencies between explicit cost, implicit cost, and energy consumption. To validate the model, we constructed a scenario, in which a medium-sized company has to decide on a software licensing model. The scenario and its parameters were populated with data about CRM solutions of Salesforce.com and Microsoft, the SME definition of the European Commission (2003), and case studies (Lai et al., 1999; Chau, 1995; Trocchia and Janda, 2003; SAManage, 2008).

The remainder of the paper is organized as follows: Our model and its criteria are introduced in Section 2. Section 3 illustrates the workings of the model and reports the validation results about the scenario constructed. Finally, Section 4 concludes the paper with a summary and a policy recommendation.

2 Proposed Model for Software Resource Management

2.1 Decision Hierarchy

The Software Licensing Selection Support (SL2S) model is based on the construction of a decision hierarchy, as suggested by AHP (Saaty, 1980). Our envisioned SL2S decision hierarchy, which is shown in Figure 1, includes all the decision criteria for selecting the appropriate software licensing model.

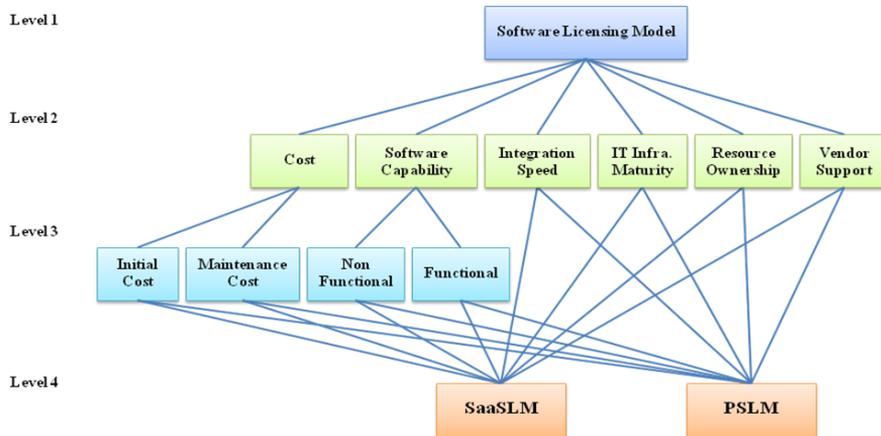


Figure 1: Software Licensing Selection Decision Hierarchy

As Figure 1 shows, SL2S model consists of 4 levels and classifies all decision criteria into 2 levels of the total four levels:

Level 1: The goal that SL2S tries to reach. The goal, which is represented as the root node of the hierarchy, is the selection of the optimal software licensing model for a given application.

Level 2: The criteria, which influence the decision making. At this level, six cost criteria can be distinguished: *Cost* for explicit cost as well as *Software Capability*, *Integration Speed*, *IT Infrastructure Maturity*, *Resource Ownership*, and *Vendor Support* for implicit cost. They are represented as the children of the root node.

Level 3: The sub-criteria, which detail criteria at level 2. This level contains four sub-criteria:

Initial Cost and *Maintenance Cost* criteria detail the criterion *Cost* and are represented as children of the *Cost* node.

Non-Functional and *Functional* criteria detail the criterion *Software Capability* with the same as its parent node.

Level 4: The alternatives that decision makers can choose from. The two alternatives, *SaaSMLM* and *PSLM*, are represented as leaf nodes of the hierarchy.

2.2 Decision Criteria

This section describes the selection criteria (representing explicit and implicit costs) that influence a company's selection of a software licensing model. While all explicit costs are defined in subsection cost, the implicit costs are inherent to all the remaining criteria: *Software Capability*, *Integration Speed*, *IT Infrastructure Maturity*, *Resource Ownership*, and *Vendor Support*. As Lai, Trueblood, and Wong (1999) already stated, the cost of alternatives must be evaluated cautiously, including their explicit and implicit costs, in order to get an objective result.

Criterion: Cost

This is a criterion, which has been considered by CIOs, who are often driven by economic objectives. The explicit cost is referred to as *total cost* (TC) of the software. The TC function is defined as:

$$TC = C_{initial} + C_{maintenance} \quad (1)$$

where $C_{initial}$ is a one-time, *initial cost* and $C_{maintenance}$ is the *maintenance cost* during the usage period of the software. (Note, with respect to general cost aspects of Green IT, besides energy, TC should also include cost of disposal and recycling.)

The initial cost refers to the cost that the organization has to pay when purchasing software. It includes server purchases (C_{hw}), software licensing cost for each client (C_{clc}) and servers (C_{slc}), network infrastructure setup cost (C_{nw}), deployment cost (C_{dep}), training cost (C_{train}), and upgrade cost (C_{up}). The initial cost is defined as:

$$C_{initial} = C_{hw} + C_{slc} + (C_{clc} \times N_{user}) + C_{nw} + C_{dep} + C_{train} + C_{up} \\ C_{initial} = C_{hw} + C_{slc} + (C_{clc} \times N_{user}) + C_{nw} + (N_{IT} \times R_{IT} \times T_{dep}) + C_{train} + C_{up} \quad (2)$$

where N_{user} is the number of users (clients), N_{IT} is number of IT technicians needed for deployment, T_{dep} is the time (in hours) needed for deploying the software per IT technician, and R_{IT} is hourly wage rate for IT technicians. Maintenance cost refers to the costs for keeping the software operating smoothly (including hardware, software, and network). It also includes the cost for using on-demand software (i.e. subscription fee). Maintenance cost is defined as:

$$C_{maintenance} = T_{use} (C_{hwm} + C_{swm} + C_{nwm} + (N_{user} \times R_{use})) \quad (3)$$

where T_{use} is the software usage duration, C_{hwm} is the hardware maintenance cost, C_{swm} is the software maintenance cost, C_{nwm} is the network maintenance cost, N_{user} is number of users, and R_{use} is the cost that the organization faces if using software on a pay-per-use basis (e.g. per month, per year, per transaction).

Criterion: Software Capability

Musa (1987) defined software capability as how well the program functions operate to meet the operational requirements. Parasuraman (1985) introduced a service quality model and identified 10 determinants of service quality (reliability, responsiveness, competence, access, courtesy, communication, credibility, security, understanding, and tangibles) as perceived by customers. Trocchia and Janda (2003) conducted interviews with Internet shoppers and identified 5 dimensions of Internet service quality (performance, access, security, sensation, and information). Based on this work, we further sub-divided software capacity into two sub-criteria: non-functional requirements and functional requirements.

Non-functional requirements refer to the software capability that is not related to the main functions of the software. These requirements include: response time, security, throughput, accessibility, availability, compatibility, software design, and reliability. Non-functional requirements also include non-technical factors, such as the learning curve to master the use of software, and the convenience of using the

software. Functional requirements refer to the usefulness of software, i.e. the functions that solve the business problem without errors. Functional requirements also include completeness and information accuracy that is generated by the software.

To evaluate these criteria, organizations need to create a check list of non-functional and functional requirements for the software. To address the Green IT objective, choosing the software with an efficient design and architecture with respect to a low number of program statements executed is important. Additionally, organization should considering software with power saving features (e.g. low-radius-color user interface) for reducing energy consumption at the client side.

Criterion: Integration Speed

Integration speed refers to the effort for integrating the new software with existing systems and finding acceptance among employees. Companies benefit from quick integration, allowing them to utilize in-house resources and concentrate on their core businesses (Ekanayaka et al., 2003). Software should not affect the progress of company projects negatively (Wohlin and Aurum, 2006). The basis for evaluating this factor is the faster the integration the better the performance. The longer the integration takes, the more disturbances are caused to the company.

Criterion: IT Infrastructure Maturity

The IT infrastructure maturity refers to the maturity level of an organization with respect to its capability in adopting a software service model. Since, from a technical perspective, software compatibility and in-house expertise are the factors affecting the selection of software package (Chau, 1995), Xin and Levina (2008) suggested that firms with a mature IT architecture are more likely to have developed standard interfaces so that they can integrate external services (i.e. SaaS).

To evaluate this factor, we consider a company's IT expertise, culture, legacy system compatibility, hardware infrastructure, and network infrastructure. The resulting list of in-house IT requirements is compared with the requirements of a SaaS or a perpetual software solution.

Criterion: Resource Ownership

Resource ownership refers to a company's situation with respect to how much resources (i.e. server, data, and software) have been outsourced. Small and medium-sized enterprises (SMEs) with limited resources (i.e. budget and IT expertise) tend to adopt IT outsourcing in order to achieve low cost of IT resource ownership (Currie, 2003). However, companies are concerned about transferring sensitive information to hosted environments, where other companies also outsource their resources to. This leads to the preference of outsourcing "non-core" software and data (Ekanayaka et al., 2003).

To evaluate this factor, decision makers of companies have to consider the trade-off between resource ownership cost and control over a certain activity or data set. The trade-off of keeping IT resources within a company is the increased maintenance cost. This means if an organization prioritizes data security over the convenience of effortless system maintenance, it should consider choosing PSLM. However, with respect to Green IT, i.e. reducing the energy consumption of servers, shifting server maintenance to SaaS providers is a good alternative.

Criterion: Vendor Support

Chau (1995) identified vendor support as an implicit cost factor, affecting software licensing selection. Chau included existing technical skills and experience of using products developed by the same vendor. On a non-technical side, Chau considered vendor reputation, business skills, references, and past business experience with the vendor. Overall, these factors impact the trust of an organization towards the software vendor with respect to fulfilling a service level contract (SLA).

In order to evaluate the factor vendor support, the organization needs to compare the services provided by the vendor and the information about the services as stated in the SLA. The criterion for evaluating vendor services includes version upgrading, customization, system maintenance, helpdesk support, and disaster recovery. Following a Green IT policy, organizations should also consider whether a vendor applied Green IT policies to their services and products. For example, a vendor is only selected if the vendor's products lower energy consumption, the vendor implemented environment-friendly data centers, or the vendor is willing to cooperate with customers to implement Green IT policies.

3 Model Validation and Result Analysis

After shortly describing the data collection, we create and analyze a scenario, in which a company has to select either the SaaS software licensing model (SaaS LM) or the perpetual software licensing model (PSLM), using the SL2S model.

3.1 Data Collection

The data used for comparing on-demand and on-premise software products is based on the software specifications published in the product catalogs of Salesforce.com and in the Microsoft CRM Web catalog. For the SaaS product, we chose the Salesforce CRM Professional Edition as the sample product. For the traditional model, we selected the Microsoft CRM on-premise solution. The quantitative data (e.g. pay-per-use rate or licensing cost) found is used for calculating the values of tangible criteria (i.e. initial cost and maintenance cost). The data about

software features and SLAs is used for comparing intangible criteria (e.g. vendor support).

3.2 Company Profile and Its Software Requirements

The company profile is built using the data found in the European Commission's SME guide (2003). It provides details about the number of annual work units and the annual turnover of small and medium-sized enterprises. In this scenario, we assume a medium-sized company that employs 100 people. Its IT budget is estimated to be in the range of 10% of its annual revenue of \$10,000,000. The profit target is set to 10% of the annual revenue. The hourly wage for 3 IT technicians is based on the rate listed at the PayScale.com Web site (PayScale, 2009).

In order to make a decision using the SL2S model, we need to set the values of the criteria. For this purpose, we base our values on values found in case studies described in literature (Lai et al., 1999; Chau, 1995; Trocchia and Janda, 2003; SA-Manage, 2008). The values used are shown in Table 1.

Table 1: Company's Software Requirements

<i>Criteria</i>	<i>Description</i>
Cost	The initial cost should not be higher than \$1 Million and maintenance costs should be less than \$500,000/year. The detailed cost analysis is shown in Table 4.
Software Capability	The new software system should be easy-to-use, should require a low learning curve, and should create high values to users.
Integration Speed	The integration time is expected to be less than 1 week.
IT Infrastructure Maturity	The company has a reliable network infrastructure. The well-established server room hosts all applications used within the organization and is scalable.
Resource Ownership	The company only trusts a highly reliable vendor to store its data, otherwise it prefers to store data under its own control.
IT Investment Policy	The organization policy is to operate any software system for 3 years before replacing it.
Usage Duration	The usage of the software is expected to be 21 months out of 3 years (i.e. 7 months per year).
Number of Users	The number of users of the software is 20.
Vendor Support	The organization heavily relies on vendor support for hardware maintenance, staff training, and helpdesk service.

3.3 Calculation of the SL2S Model Values

As a first step, the medium-sized company, which we have created in the previous section, has to perform pairwise comparisons between all criteria. The results are illustrated in the Criteria Judgment Matrix (Table 2). The criteria judgments are based on assumptions found in Currie (2003), Ekanayaka(2003), as well as in Trocchia and Janda (2003). Note, applying our SL2S model at a real company in practice, the judgments need to be made by a committee consisting of experts

from the financial, managerial, and technical departments of the company. Although not shown here, pairwise comparisons of all sub-criteria (i.e. for initial cost and maintenance cost as well as for non-functional and functional) need to be performed as well.

Table 2: Criteria Judgment Matrix of the SL2S Model

<i>Criteria</i>	<i>COST</i>	<i>SWCAP</i>	<i>INTSPD</i>	<i>INFMAT</i>	<i>RESOWN</i>	<i>VENSUP</i>
<i>COST</i>	1	3	7	3	5	3
<i>SWCAP</i>	1/3	1	5	3	2	2
<i>DEPSPD</i>	1/7	1/5	1	1/3	1/5	1/7
<i>INFMAT</i>	1/3	1/3	3	1	1/4	1/5
<i>RESOWN</i>	1/5	1/2	5	4	1	1/2
<i>VENDUP</i>	1/3	1/2	7	5	2	1

Where *COST* is cost, *SWCAP* is software capability, *INTSPD* is integration speed, *INFMAT* is IT infrastructure maturity, *RESOWN* is resource ownership, and *VENSUP* is vendor support.

To check that the judgments made in Table 2 are consistent, the consistency ratio (CR) method can be applied (for details see Saaty, 1980). The CR value for our data is 0.09, which is acceptable (Karlsson, 1997). As a next step, following the AHP approach, a normalized matrix has to be derived by dividing each matrix entry of Table 2 by the sum of its column. Since this is a simple step to perform, those normalized matrices are not shown. In a subsequent step, we calculate the average of each row, i.e. for each criterion. The result is the relative weight for each criterion (Table 3). The same has to be applied to the sub-criteria judgment matrices, resulting in 2/3 to 1/3 for cost and in 1/2 to 1/2 for software capability.

Table 3: Relative Weights of the Criteria of the SL2S Model

<i>COST</i>	<i>SWCAP</i>	<i>INTSPD</i>	<i>INFMAT</i>	<i>RESOWN</i>	<i>VENSUP</i>
0.3866	0.1948	0.0322	0.0707	0.1280	0.1877

3.4 Comparison of the SaaSMLM and PSLM

As a next step, pairwise comparisons between each criterion of the two software licensing alternatives (SaaSMLM and PSLM) have to be performed. For this, first, the cost input factors are extracted from the CRM software specification of Salesforce's SaaS solution and Microsoft's on-premise solution. Table 4 shows all those cost input factors for both licensing models.

Table 4: Cost Input Factors

Category	Hardware Cost	Server License Cost	Client License Cost (per User)	Deployment Cost	Training Cost	Upgrade Cost	Hardware Maintenance Cost (per Month)
SaaS/SLM	\$0	\$0	\$0	40h	\$50	\$0	\$0
PSLM	\$600	\$2,000	\$1,000	180h	\$300	\$0	\$5

Category	Software Maintenance Cost (per Month)	Network Maintenance Cost (per Month)	Subscription Rate (per Month)
SaaS/SLM	\$0	\$10	\$65
PSLM	\$244	\$10	\$0

In order to evaluate the total cost factor, we calculate the initial cost and maintenance cost for the SaaS/SLM model and for the PSLM model using equation (2) and equation (3), respectively. The results are shown under *INITCOST* and *MAINCOST* in Table 5.

Table 5: Alternative Judgment Table

Model	<i>INITCOST</i>	<i>MAINCOST</i>	<i>NONFUNC</i>	<i>FUNC</i>	<i>INTSPD</i>	<i>INFMAT</i>	<i>RESOWN</i>	<i>VENSUP</i>
SaaS/SLM	500	25,900	1	1	4.5	1	1	1
PSLM	27,510	5,439	1	1	1	5	5	1

Where *INITCOST* represents the initial cost, *MAINCOST* represents the maintenance cost, *NONFUNC* represents the non-functional software capabilities, and *FUNC* represents the functions of the software.

Table 5 also shows that there is no difference between SaaS/SLM and PSLM with respect to the non-functional (*NONFUNC*) and functional (*FUNC*) software capabilities. This means that all the requirements of the medium-sized company can be fulfilled by both software alternatives independent of its licensing type. The integration speed (*INTSPD*) is assumed to be better for the SaaS/SLM model, based on the vision of SaaS that business processes and their workflows of activities can be changed quickly. However, it has to be noted that the integration speed might be equal for companies which just started to use software services. A similar reasoning is valid for the infrastructure maturity (*INFMAT*) criterion. Over time, the IT infrastructure of companies will become more mature, making the values for the PSLM closer to 1. Currently, the existing IT management infrastructures need more effort to integrate SaaS models. The resource ownership (*RESOWN*) criterion is still an issue currently. However, over time, improved technology and positive experience with SaaS providers will reduce the current advantage of the PSLM model. Finally, the vendor support (*VENSUP*) criterion evaluates the performance and reputation of providers. In our scenario, since we assume that the SaaS offer and PSLM offer come from the same vendor, both values are set to 1.

Based on the values in Table 5, we calculate the ratio for each of the criteria for SaaS/SLM and for PSLM. For example, the ratio for the initial cost factor is $500/(500+27510) = 0.0179$ for PSLM and $27510/(500+27510) = 0.9821$ for SaaS/SLM (note, for cost only, the values are reversed). Then, these weights are mul-

multiplied with the relative weight for this criterion (i.e. $0.3866 * 2/3$), as listed in Table 3. By applying the same procedure to the other criteria, Table 6 is obtained.

Table 6: Comparison of Both Alternatives

Alternatives	COST		SWCAP		INTSPD	INFMAT	RESOWN	VENSUP	Total
	INIT COST	MAIN COST	NONFUNC	FUNC					
SaaS _{LM}	0.2531	0.0224	0.0487	0.0487	0.0264	0.0118	0.0213	0.0939	0.5262
PS _{LM}	0.0046	0.1065	0.0487	0.0487	0.0059	0.0589	0.1067	0.0939	0.4738

Based on the results in Table 6, we can see that the sum of these values is higher for SaaS_{LM} than for PS_{LM}. This implies that the SaaS licensing model is the most appropriate solution under our scenario for the medium-sized company.

3.5 Modeling Uncertainty of Usage Duration

When populating the criteria of the SL2S model, there is uncertainty about the correctness of the values. This uncertainty is simply based on the fact that any prediction can be wrong. For instance, there is a high uncertainty about the usage duration. If a company uses SaaS longer than predicted, then this company has to pay higher subscription fees than initially expected. Consequently, because of the higher cost, the SaaS licensing model becomes less preferred than the perpetual software licensing alternative. Moreover, there is the likelihood that the company’s decision on choosing SaaS had not been the correct one.

In order to address this issue, we analyze the impact of the usage duration on the final decision. For this purpose, we execute our implementation of the SL2S model for usage duration values from 1 to 36, keeping all other values of the criteria fixed (except for the maintenance cost that is based on the usage duration). The outcome of this sensitivity analysis is illustrated in Figure 2.

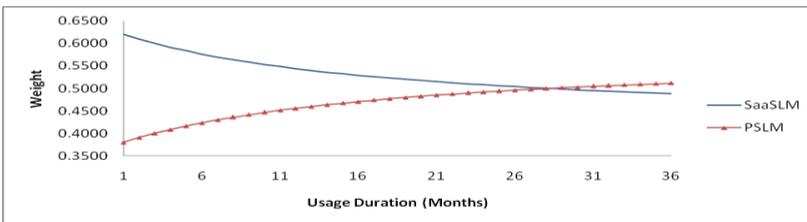


Figure 2: Effect of Usage Duration on the Decision Making

Figure 2 illustrates that SaaS_{LM} is better than PS_{LM} up to 28 months of usage duration. It also shows that SaaS_{LM} becomes less preferred with increasing usage duration until it is equal as the PS_{LM} in month 28 (intersection point of both curves). If we compare the total cost (direct cost) of SaaS_{LM} and PS_{LM}, we get a similar figure (Figure 3). Because of the close relationship between energy

consumption and direct cost, this figure also indicates the preferred decisions with respect to energy conservation.

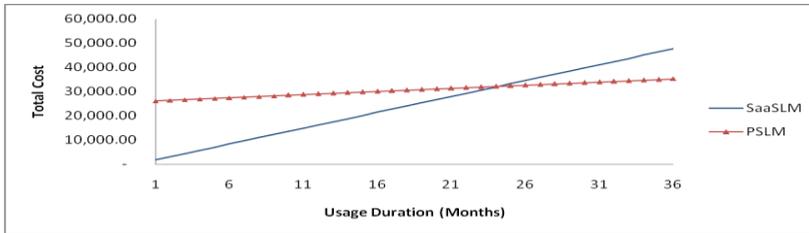


Figure 3: Relationship of Total Cost and Usage Duration

Figure 3 also illustrates that, since the SaaS model incurs less initial cost, SaaS is more preferred to customer with low usage duration. This situation, however, changes at the equilibrium point of SaaS and PaaS, which is at month 24.

Comparing the intersection points of Figure 2 and Figure 3, it is clear that the equilibrium points are close to each other but different. The reason is that the outcome from the SL2S decision making model also depends on the other model criteria not only on the total cost (i.e. explicit cost). In order to understand the impact of cost on the overall decision, we analyze the relationship between weights of the (explicit) cost of SaaS, the (explicit) cost of PaaS, the other (implicit cost) criteria of SaaS and the other (implicit cost) criteria of PaaS (Figure 4).

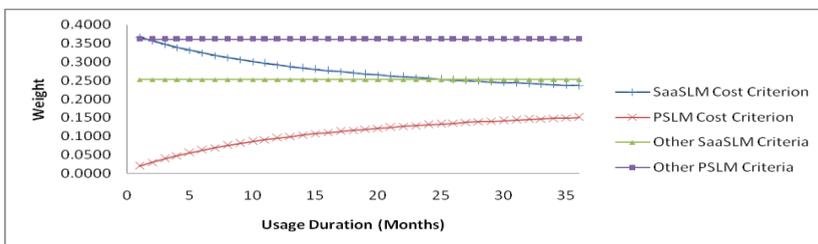


Figure 4: Impact of Cost Factors on the Selection Decision

Figure 4 depicts that the implicit cost criteria contribute 25% (SaaS) and 36% (PaaS) with respect to the overall decision. Since this is substantial, any Green IT policy has to consider not only explicit cost but also its relationship to the other (implicit cost) criteria. For example, if a company decides to prioritize Green IT, it has to re-evaluate the relative weights of the implicit cost criteria.

4 Conclusion and Policy Recommendation

In this paper, we proposed a decision making model for selecting the software licensing model that is most appropriate to a company. This model is called Software Licensing Selection Support (SL2S) Model. The Analytic Hierarchy Process

(AHP) method has been used as a basis for our decision making model. The model criteria, which influence the software licensing model selection, are cost, software capability, integration speed, IT infrastructure maturity, resource ownership, and vendor support. The software licensing models that we compared are the software-as-a-service licensing (SaaS) model and the perpetual software licensing (PSL) model. The SL2S model also allows performing sensitivity analyses and provides guidelines on how to assess and relate those criteria.

Finally, our model has been validated by applying it to a constructed scenario, in which a medium-sized company has to decide on a software licensing model. In our analysis, we investigated not only the overall decision outcome under explicit cost but also its relationship to implicit cost criteria and energy consumption, as needed for establishing a Green IT policy. The results show that the implicit cost criteria (25% of SaaS) are a major decision factor. This suggests that a sustainable IT policy cannot only consider lowering cost (as a mean for lowering energy consumption) but also has to re-evaluate its relation to implicit cost criteria.

References

- Chau PYK (1995) Factors used in the selection of packaged software in small businesses: Views of owners and managers. In: *Information and Management* 29:71-78.
- Choudhary V (2007) Comparison of Software Quality Under Perpetual Licensing and Software as a Service. In: *Management Information Systems* 24(2):141-165.
- Currie WL (2003) A Knowledge-based Risk Assessment Framework for Evaluating Web-enabled Outsourcing Projects. *Internatl. Journal of Project Management* 21:207-217.
- Ekanayaka Y, Currie WL, and Seltsikas P (2003) Evaluating application services providers. In: *Benchmarking: An International Journal* 10(4):343-354.
- European Commission. (2003) *The new SME definition: user guide and model declaration.*
- Ghodsypour SH, and O'Brien C (1998) A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming. In: *International Journal of Production Economics* 56-57:199-212.
- Jadhav AS, and Sonar RM (2009) Evaluating and selecting software packages: A review. In: *Information and Software Technology* 51:555-563.
- Kaplan JM (2007) SaaS: friend or foe? In: *Business Communications Review* 2007:48-53.

- Karlsson J (1997) A Cost-Value Approach for Prioritizing Requirements. In: IEEE Software September/October:67-74.
- Lai VS, Trueblood RP, and Wong BK (1999) Software selection: a case study of the application of the analytical hierarchical process to the selection of a multimedia authoring system. In: Information & Management 36:221-232.
- Musa J (1987) Software Quality and Reliability Basics. In: Fall Joint Conference IEEE Exploring Technology Today and Tomorrow.
- Parasuraman A, Zeithaml VA, and Berry LL (1985) A Conceptual Model of Service Quality and its Implications for Future Research. In: Journal of Marketing 49(4):41-50.
- PayScale (2009) Hourly Rate Chart for Computer/Network Technicians. www.PayScale.com. Fetched on 2009-02-10.
- Risch M, and Altmann J (2008) Cost Analysis of Current Grids and Its Implications for Future Grid Markets. In: Lecture Notes in Computer Science 5206:13-27.
- Saaty TL (1980) The Analytic Hierarchy Process. McGraw-Hill, New York.
- SAManage (2008) SaaS vs. Traditional ITAM Tools: How the Hosted Approach Delivers Greater Value. www.SAManage.com. Fetched on 2008-05-06
- SIIA (2006) Software as a Service: A Comprehensive Look at the Total Cost of Ownership of Software Applications. <http://www.siia.net>. Fetched on 2009-11-25.
- Trocchia PJ, and Janda S (2003) How do customers evaluate Internet retail service quality? In: Journal of Services Marketing 17(3):243-253.
- THINKStrategies in conjunction with Cutter Consortium. (2008) What is the Greatest Benefit of the SaaS Offering? www.thinkstrategies.com. Fetched on 2008-10-12.
- Wang G, Huang SH, and Dismukes JP (2004) Product-driven supply chain selection using integrated multi-criteria decision-making methodology. In: International Journal of Production Economics 91:1-15.
- Wohlin C, and Aurum A (2006) Criteria for selecting software requirements to create product value: An industrial empirical study. In: Value-Based SW Engineering. Berlin.
- Xin M, and Levina N (2008) Software-as-a-Service Model: Elaborating Client-side Adoption Factors. In: Proc. of 29th Intl. Conf. on Information Systems, Paris, France.