Comparison of different revenue models in SaaS

Arto Ojala
Department of Computer Science and Information Systems
University of Jyväskylä
Jyväskylä, Finland
arto.k.ojala@jyu.fi

Abstract —**Cloud computing brings new possibilities for software firms to sell their products within a Software-as-a-Service (SaaS) model. However, although SaaS provides new revenue models, it may not easily achieve a profitable revenue stream. For customers, SaaS promises cost savings related to IT expenses. However, it may be difficult to estimate which revenue model will be best for a given situation. In this paper I present the common revenue models used in SaaS, examining their advantages and disadvantages from the point of view of the software provider and the customer.**

Keywords-Revenue models, SaaS, Cloud computing, Business in cloud

I. INTRODUCTION

Cloud computing is changing the way in which software is sold, delivered, and used. In SaaS model, software is delivered as a service to customers via the Internet. For customers, SaaS provides online access to software when required, without the need to have it permanently installed on their own computers. It also ensures that the latest version of the software is in use without having to install continuous updates. In addition, because the software is executed on a service provider's server, users no longer have to worry about the technical specification of their computer, or about the data storage capacity.

However, the advantages are not all on one side. The software vendor will have to ensure a profitable revenue stream when an initial license fee is replaced by a usage-based fee or a rental fee. And for the customer, SaaS may increase uncertainties related to data security when important data is stored on a public cloud server, and when applications are dependent on the reliability of the Internet connection.

The revenue models in SaaS are mainly based on the actual usage of the software over the Internet (pay-per-use), or on software rental [1, 2, 3]. Pay-per-use involves charging the customer only for the metered usage of the software, irrespective of the overall period for which the software is used. In software rental, the customer pays a negotiated subscription fee for a certain time period. In some cases a firm may also offer a traditional licensing model in addition to its SaaS offering. In this case the customer can have the option of buying a traditional license for a workstation, or can use the software within a private cloud.

In this article I will give an overview of the three main revenue models. I will look at their advantages and disadvantages for the SaaS providers and their customers. Thus, the focus is on SaaS providers and their customers, not on Infrastructure-as-a-Service (IaaS) or Platform-as-a-Service (PaaS), in which the revenue models may differ.

II. CLOUD COMPUTING

Cloud computing refers to the provision of computing capacity, storage capacity, and applications as a service over the Internet. In line with Armbrust et al. [1], I consider cloud computing to cover not just software applications delivered through the Internet, but also the hardware and system software that data centers use to provide these services. The three service layers in cloud computing are IaaS, which provides computation and storage capacity, PaaS, which provides software development tools plus an application execution environment, and SaaS, which provides applications on top of a PaaS, IaaS, or private data center [1].

The data center hardware and software that form a "cloud" can be divided into a public cloud, a private cloud, and a hybrid cloud. In a public cloud, software vendors use their own or a third party's cloud infrastructure (data center) to offer SaaS for customers on a pay-as-you-go basis. A private cloud involves the customer's own internal data center, where the software is installed and used in a centralized manner within the organization – in other words, the software is not made publicly available [1, 4]. In the case of a hybrid cloud, a firm using a private cloud may, for example, offload part of its workload to a public cloud and in that way acquire more computing capacity [4, 5].

III. REVENUE MODELS IN BRIEF

Since software belongs to the category of information goods that can be delivered and sold either in a material form (e.g. as a CD or a DVD package), or in an intangible form through the Internet, it can be used with several revenue models. Software products usually have a very high initial cost, due to the substantial human resources required for product design and development. However, for a ready-made software product, the reproduction costs can be practically non-existent. This makes the pricing strategies of software products somewhat different from those in other industries [6, 7].

Several terms are used (mostly as synonyms) to describe how a firm makes money by selling its products or services. These terms include, for example: *a business model, revenue logic, a pricing model, earning logic, a licensing model*, and *a revenue model*. The term *revenue model* is here used in an operational sense, referring to how a firm collects revenue from its customers. Thus, it relates to the various options that a firm

may offer to customers who might wish to buy its software [11].

The software can be also sold using several revenue models, or combinations of different models. These models would include packaged and server-based licensing, software renting, pay-per-use pricing, effort-based pricing (e.g. in software projects), revenue sharing with partners, utility-based charging, freemium, advertisement-based models, and so on [10, 11]. Here I will focus on the most common revenue models, namely software renting and pay-per-use, which can be provided by SaaS. I will also discuss the advantages and disadvantages of traditional software licensing in comparison to pay-per-use and software renting models (Table 1). An assumption here is that the SaaS providers sell the software using a public cloud, although in some cases the SaaS providers may rent the software for use in a customer's private cloud.

IV. PAY-PER-USE

Pay-per-use involves charging the customer according to the metered usage of the software, that is, how much use is made of software. Thus, there is a unit with a fixed price and a customer will be charged periodically, based on the units used. The metered unit may be based on the length of time that the software is running, the number of times that a key subroutine is called, the number of transactions handled, or some combination of these [12].

For software vendors, the pay-per-use model makes it possible to diversify their customer base. In other words, software can be available to smaller customers who may not have sufficient financial resources to buy a traditional software license. The model is also suitable when the customer needs the software only occasionally, for some specific purpose. The model also promotes the *network externalities effect*, meaning that the increased number of buyers increases knowledge of the product among potential customers, lowers customers' search costs, and makes the product well known in the market [8]. For a software vendor, it is hard to know how many copies are made of software, but easy to know how many times it is used. This situation favors the pay-per-use model [8], since when software is executed on a cloud server, software piracy becomes practically impossible [13, 14].

The disadvantages of the pay-per-use model for software vendors are related to the customers' low switching costs, maintenance of auditable records of usage, and the risk of not covering the high development costs of the software. Since customers do not make long-term contracts for use of the software, switching costs are very low, and customers may change the software provider if there are alternative applications available at a lower price. If the pricing is based on usage, the software vendor may need to maintain and, on demand, present an itemized document of each customer's usage of the software [9]. This increases administrative work and the need for a reliable process to deliver these documents. The development of the software is very costly, because of the resources needed to design and develop the software product. In the pay-per-use model, where initial incomes are very low and uncertain, recouping the development costs is bound to be more risky than with traditional licensing.

Especially for smaller customers that do not have a budget for the kind of investments required by traditional software licensing, the pay-per-use model is a good option. Customers can purchase the software without having to make special budgeting arrangements or undertake long decision-making processes. Thus, usage of the software resembles an operational cost rather than a capital investment. The pay-per-use model also has advantages over other models if customers need the software only occasionally, since this decreases costs, and lessens the need for customers to set up their own IT infrastructure. As mentioned above, switching costs are very low and it is easy to change a software vendor if the quality/functionality of the software is not at the appropriate level. Thus, in this model it is possible to test and evaluate the suitability of the software for the company's needs. It also frees the customer from the need to install, maintain, and update the software, and it lessens the need to have ones own IT personnel. All in all, this means that there are no hidden costs related to software usage – bearing in mind that hidden costs in traditional software licensing can increase a firm's IT budget by as much as 80 percent [3].

From the customer point of view, the disadvantages of the pay-per-use method are related to software pricing, estimation of the actual need to use the software, data security, and the permanence of the software provider in the market. In most cases, the price of the software offered by the pay-per-use method is like a list price – the fee will be just the same for all the customers using the software in a cloud. This makes it impractical to have price negotiations with the software vendor. In addition, it is often difficult to estimate in advance how much the software will actually be used. Thus, so long as the software is available through other revenue models, these models may be more appropriate if the software is needed a continuous basis. One further concern may involve data security. If a firm's data is stored on a public cloud, it may not know precisely where the data is stored, and there may well be other data sources that are stored collectively, along with the firm's data. Finally, for customers who use the software for their core business, the continued existence of the cloud provider (providing SaaS, IaaS, or PaaS) is of vital importance. If a cloud provider disappears from the market, the effect on the customer will be immediate, and possibly catastrophic.

V. SOFTWARE RENTAL

In software rental, the customer pays a negotiated subscription fee to use the software license for a certain limited time period [1]. In cloud business, software rental shares several similarities with the pay-per-use model.

For the vendor, software renting as a cloud service offers more pricing flexibility than the pay-per-use model. The price of the rental can be based on the length of the agreement, the number of users in the customer organization, the functionalities of the software, or the size of the customer. Thus, the software may be offered more cheaply to smaller customers than to large corporations. The rental model is also

simpler to apply than the pay-per-use method, which requires metering of actual software usage. This also means that if the software vendor uses a third party's IaaS and/or PaaS service to provide the SaaS offering, the software can be priced independently. The third party's costs related to the use of IaaS and PaaS can then be added. In the long run, the rental model may generate more revenue than the other models, so long as the vendor can maintain loyal customer relationships. The disadvantages of software renting for the vendor are fairly similar to those in the pay-per-use model (see Table 1), excluding the need for maintain auditable records of software usage.

For customers, software renting offers more possibilities for price negotiations, and for varying the terms of usage and length of the contract. The total costs of the software are also predictable and contractually defined. Hence, there are no hidden costs related to the software, and the customer knows the financial resources that need to be allocated during the contract. Once again, the disadvantages are rather similar to those for the pay-per-use model discussed above. The major difference is that in the rental model, the customer pays, irrespective of whether or not the software is actually used.

VI. SOFTWARE LICENSING

In traditional software revenue models, software is commonly sold as packaged or server-based licensing. In packaged licensing, a customer buys a single license for a single user or computer, whereas in server-based licensing, the software is bought for a certain number of processors running the software. Hence, the software can only be run on a limited number of computers. In addition to the initial license fee, customers often have to pay a maintenance fee, which includes technical support and version updates [10].

In traditional licensing, the benefits for a software provider come from high license fees. These help to cover the development costs of the software, and they do so in a shorter time period than with other revenue models. A high license fee also increases switching costs for the customer; thus, if the software is appropriate, it increases customers' loyalty. However, there is little income from the customers afterwards. In addition, because the software is installed in the customer's premises, it makes misuse of the license or direct software piracy more likely.

From the customer's point of view, traditional licensing can have advantages over other models if the software is needed for a long period of time, and if it is for everyday use in the firm's core business. In these cases, buying the license could be more profitable than rental or pay-per-use models. Buying the license also makes it possible to store and secure the data within the firm's own data center.

The disadvantages for customers in the case of licensing include hidden costs related to various aspects including maintenance fees, installation and configuration, hardware, and employment of IT personnel. Thus, the purchase price is often just a small part of the total costs of the software [3]. Buying the software can be a major investment for a firm, requiring budgeting and decision-making at top management level. Time is also needed for planning and implementation. In addition, switching costs are high and the model tends to tie the customer to the software selected. In traditional licensing, a company also needs its own IT infrastructure, that is, data storage and computing capacity.

VII. CONCLUSIONS

As discussed above, cloud computing brings both possibilities and challenges to software providers and customers. In some cases, traditional software licensing still has advantages over the SaaS model. However, from the arguments presented in this paper, one can also see the benefits of software rental, in that it can provide a trade-off between traditional licensing and the pay-per-use model. Software rental can suit both large and small customers, and both wide and narrow target segments; it further helps to protect against software piracy, and provides a predictable and less risky revenue stream. However, it is not the most advantageous model in every case. Altogether, software providers may increase profitability and expand their customer base by providing (1) a traditional license for large customers who make extensive use of the software in their core business, (2) a rental model for mid-class users, and (3) a pay-per-use model for occasional users.

From the customer's point of view, too, software renting can provide a trade-off between traditional licensing and the pay-per-use model. It has the particular advantage of making it possible to estimate the costs of the software, and to buy the software without separate budgeting or complicated decision processes.

ACKNOWLEDGMENT

The research reported in this paper was carried out within the framework of the Cloud Software Program which was governed by TIVIT Oy nominated to organize and manage the programs of the Strategic Center for Science, Technology and Innovation in the field of ICT funded by the Finnish Funding Agency for Technology and Innovation (TEKES). In addition, the author would like to thank financial support from *Finnish Foundation for Economic Education* for this study.

REFERENCES

[1] M. Armbrust et al., "A view of cloud computing," *Communication of the ACM,* vol. 53 no. 4, 2010, pp. 50-58.

[2] V. Choudhary, "Comparison of Software Quality Under Perpetual Licensing and Software as a Service," *Journal of Management Information Systems*, vol. 24, no. 2, 2007, pp. 141-165.

[3] B. Waters, "Software as a service: A look at the customer benefits," *Journal of Digital Asset Management*, vol. 1, no. 1, 2005, 32-39.

[4] P. Louridas, "Up in the Air: Moving Your Applications to the Cloud," *IEEE Software*, vol. 27, no. 4, 2010, pp. 6-11.

[5] Sotomayor et al. "Virtual Infrastructure Management in Private and Hybrid Clouds," *IEEE Internet Computing*, vol. 13, no. 5, 2009, 14-22.

[6] Y. Bakos and E. Brynjolfsson. "Bundling Information Goods: Pricing, Profits, and Efficiency," *Management Science*, vol. 45, no. 12, 1999, pp. 1613-1630.

[7] B. Mahadevan, "Business Models for Internet-Based E-Commerce: AN ANATOMY," *Californian Management Review*, vol. 42, no. 4, 2000, pp. 55-69.

[8] V. Choudhary et al., "Economic Benefits of Renting Software," *Journal of Organizational Computing and Electronic Commerce*, vol. 8, no. 4, 1998, pp. 277-305.

[9] A. Sundararajan, "Nonlinear Pricing of Information Goods," *Management Science*, vol. 50, no. 12, 2004, pp. 1660-1673.

[10] D. Ferrante, "Software Licensing Models: What's Out There?," *IT Professional*, vol. 8, no. 6, 2006, pp. 24-29.

[11] L-M. Sainio and E. Marjakoski, "The logic of revenue logic? Strategic and operational levels of pricing in the context of software business," *Technovation*, vol. 29, no. 5, 2009, pp. 368-378.

[12] K.D. Hunter, "Metering system with remotely resettable time lockout," United States Patent nro. 5377268, 1994.

[13] A. Ojala and P. Tyrväinen, "Developing Cloud Business Models: A Case Study on Cloud Gaming," *IEEE Software*, vol. 28, no. 4, 2011, pp. 42-47.

[14] A. Ojala and P. Tyrväinen, "Value networks in cloud computing", *Journal of Business Strategy*, vol. 32, no. 6, 2011, pp. 40-49.

TABLE I. COMPARISON OF ADVANTAGES AND DISADVANTAGES FOR SAAS REVENUE MODELS AND SOFTWARE LICENSING

	Software provider		**Customer**	
	Advantages	Disadvantages	Advantages	Disadvantages
Pay-per-use	-Diversification of the customer base -Network externalities effect -Makes software piracy impossible	-Risk of not recouping development costs -Need to maintain auditable records of usage -Customer has low switching costs	-No high initial investments -Suits occasional usage -Allows shift from capital investment to operational costs -Low switching costs -Possible to test and evaluate the suitability of the software -No need to install, maintain, or update -Less expense on own ICT personnel -No need for own IT infrastructure	-Same price for all customers (non-negotiable) -Data security concerns -Actual usage difficult to estimate
Software rental	-Flexible pricing strategies -Diversification of the customer base -Network externalities effect -Makes software piracy impossible -No need to meter actual usage -More profitable if customers remain loyal -Cumulative profits	-More risk of not recouping development costs -Customer has low switching costs	-Price negotiable -No high initial investment -Costs are predictable -Allows shift from capital investment to operational costs -No need for separate budgeting -Suitable when product is needed for a fixed period -Possible to test and evaluate the suitability of the software -No need to install, maintain, or update -No need for own IT infrastructure -Less expense on own ICT personnel -Low switching costs	- Costs are the same whether of not the software is used -Data security concerns -Risk of software provider falling out of the market
Software licensing	-Easier to recoup development costs -Customer has high switching costs	-No substantial income afterwards -Risk of license misuse and software piracy	-Suitable when product is needed for a long period -Suitable when software is used for the firm's core business -Price negotiable -Data stored and secured within own premises	-Requires separate budgeting and decision-making -Need for data storage and computing capacity -Hidden costs -High switching costs