Jyväskylän yliopiston julkaisuarkisto
Jyväskylä University Digital Archive

UNIVERSITY OF JYVÄSKYLÄ

This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.

Author(s): Ojala, Arto

Title: Adjusting Software Revenue and Pricing Strategies in the Era of Cloud Computing

Year: 2016

Version:

Please cite the original version:

Ojala, A. (2016). Adjusting Software Revenue and Pricing Strategies in the Era of Cloud Computing. Journal of Systems and Software, 122(December), 40-51. https://doi.org/10.1016/j.jss.2016.08.070

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Adjusting Software Revenue and Pricing Strategies in the Era of Cloud Computing

Arto Ojala
University of Jyväskylä

Abstract

Recent research has recognized cloud computing as a new paradigm of servitization in which software products are offered based on service contracts. Thus, instead of selling software licenses, software vendors can rent software as a service to customers. However, it is still unclear how software providers can use software renting as a competitive strategy in the software market. Based on 37 interviews with software professionals from five case firms, this paper focuses on the connection between competitive forces and the factors influencing the selection of a pricing model. The findings indicate that servitization of the software offering makes it possible to adjust revenue and pricing strategies relative to market competition. Depending on the competitive situation in the market, firms apply mixed revenue models, or else a hybrid pricing mechanism, to protect their business against rivalry and substitutes. The software renting model has several advantages which significantly help software vendors to expand their business opportunities. However, in some cases, powerful customers are able to limit the revenue and pricing options. The findings also indicate that software renting is related to cost leadership and differentiation strategies, whereas software licensing is linked to a focus strategy.

Keywords: software pricing, servitization, competitive strategy, cloud computing, SaaS

1. Introduction

Cloud computing is servitizing the information technology (IT) industry (Huang and Shen, 2015; Sultan, 2014a, 2014b; Wortmann et al., 2012). The increasing movement

towards servitization of the software offering through cloud computing is changing the competitive environment in the software industry (Cusumano, 2010; Sultan, 2014a, 2014b; Wortmann et al., 2012), and challenging existing business strategies (Bustinza et al., 2015). To compete in the software market, software developers have to take these changes into account and to rethink how to offer their software to customers, given that traditional software licensing[1] and fixed software development contracts are becoming more demanding (see e.g., Ahonen et al., 2015; Cusumano, 2007, 2010; Lee et al., 2013). The movement to a servitization model facilitates the offering of software through service contracts (cf. Barnett et al., 2013) rather than by licensing software to a customer. In this situation, servitization of traditional software products into a software as a service (SaaS[2]) model permits a number of revenue[3] and pricing models. However, it also involves challenges, making it difficult to sustain a profitable revenue stream.

Because software products belong to the category of intangible information goods (Valtakoski, 2015), the price of the first product is usually very high, while reproduction and delivery costs are, in many cases, close to zero. This allows software providers to use varied revenue and pricing models (e.g., Linde, 2009). In most cases, the copyright for the software belongs to the producer. Thus, the software is licensed to the customer, and the license may limit usage of the software in such a way that the licensee cannot resell, modify, or re-rent the software without the permission of the provider (Choudhary et al., 1998; Sun et al., 2008). The traditional way to sell the software license is to sell a perpetual software license for a single user or machine (out-of-the-box software), or else to sell a license to use the software in a certain number of processors (Ferrante, 2006). Recently, a number of studies have suggested that software renting is becoming more frequent in the new era of servitization, within which software is delivered via the SaaS model (Huang and Shen, 2015; Choudhary, 2007; Leavitt, 2009; Ojala, 2012, 2013; Sultan, 2014a; Waters, 2005).

[1] In this study, "software licensing" refers to a situation in which a customer buys a software license for a single user or a certain number of processors, with no time limitation on usage of the software.
[2] In line with Armbrust et al. (2010) and McAfee (2011), SaaS refers here to the software delivery model, irrespective of the revenue or business model used. In the SaaS model, software is delivered as a service and used over the Internet.
[3] The term "revenue model" is used here to refer to different ways of offering software, including software renting, software licensing, and pay-per-use methods.

A number of studies on economic models have investigated the benefits of renting as compared to buying a product outright (e.g. Bucovetsky and Chilton, 1986; Choudhary, 2007; Choudhary et al., 1998; Flath, 1980). These studies have used analytical approaches that simplify real-world settings, seeking to apply algebraically testable rules to determine the benefits of rental agreements. However, these economic models cannot be applied to the software industry as they stand, since they have been based on monopolistic market situations and have neglected market competition. For this reason, it remains unknown how software firms select their revenue and pricing models in competitive markets, where they have several options that can be applied exclusively or in parallel.

In seeking to increase our theoretical understanding of software revenue and pricing strategies, and to fill the research gaps discussed above, we applied Porter's (1980, 2008) theory of competitive advantage. Our primary goal was to address one overall question, namely: *How can SaaS providers adjust their revenue and pricing strategies in competitive markets?* This question can be divided into two sub questions: 1) *What are the competitive forces that drive SaaS providers to rent their software?* and 2) *What are the competitive forces that inhibit SaaS providers from renting their software?* Thus, the aim was to study the link between competitive forces in the market and particular revenue and pricing models. In addition, this study responded to the recent call by Smolander et al. (2016) in *Journal of Systems and Software,* urging researchers to develop a better understanding of the interactions between software business and software engineering. This is especially important in the context of cloud computing, given that – as noted by Li et al. (2016, 13) – "Appropriate pricing schemes and techniques are crucial for developing and maintaining a successful and sustainable Cloud ecosystem."

As a theoretical contribution to the field, this research first of all indicates how competitive forces (Porter, 1980) in the market shape software revenue and pricing models in cloud computing, and how different revenue and pricing models lead to different competitive strategies. Secondly, the study contributes to the emerging literature on IT servitization (Bustinza et al., 2015), by indicating how IT servitization makes it possible to use different competitive strategies flexibly, according to market competition

and customers' preferences. Thirdly, the findings here add to earlier studies on software renting and pricing, revealing how servitization is moving revenue models from selling products towards rental contracts (Barnett et al., 2013). Finally, this study adds to previous economic literature and studies on software renting (which have mainly applied predetermined variables), and extends the focus to cover monopolistic market situations. From a practical perspective, the factors impacting on software pricing are important for managers deciding on software engineering options, since the software architecture[4] that is developed may either limit or open up possibilities for different revenue and pricing models (Laatikainen and Ojala, 2014).

The paper is organized as follows: Section 2 discusses related studies on cloud computing, software revenue models, and competitive strategies. Section 3 presents the qualitative research method applied in this study, followed in Section 4 by the case study findings. Section 5 discusses the findings in relation to previous literature. Finally, Section 6 presents concluding thoughts and practical implications.

2. Related work

This section aims first to review servitization in the context of cloud computing and how it impacts on the market strategies of SaaS providers and adopters. Thereafter, the benefits of software rental are discussed from the perspective of software customers and providers, with discussion also of the different pricing models used in software renting. Finally, Porter's (1980) theory of competitive strategy is presented and discussed, in relation to the framework of the present study.

2.1. Servitization in the context of cloud computing

Servitization has been of increasing interest to scholars, starting from the late 1970s (Baines et al., 2009). A seminal paper by Vandermerwe and Rada (1988, 314) defined the

[4] Software architecture refers to the components and relationships of the internal parts of a software system (Bass et al., 1998), such as user interface components and database management systems, which may be available only in desk-top computers, traditional back-office servers, cloud platforms, or multiples of these environments.

concept as "Market packages or 'bundles' of customer-focused combinations of goods, services, support, self-service and knowledge." Since then, several alternative definitions have emerged, and servitization is now commonly used as an umbrella term for a variety of service-based models (see e.g. Baines et al., 2009; Schroeder and Kotlarsky, 2015). Traditionally, servitization has been studied in the context of manufacturing, encompassing how manufacturing firms can create value by adding services to their products (Baines et al., 2009; Sultan, 2014a). However, in the case of cloud computing, servitization is seen as a new model by which hardware and software can be turned into a service (Schroeder and Kotlarsky, 2015; Sultan, 2014a, 2014b; Wortmann et al., 2012). On the basis of these previous works, and the context of present study, servitization is seen as a process whereby the physical IT environment and traditional software products are transformed into a service that is offered for customers over the Internet. In a servitized software offering, customers do not need to have their own physical IT infrastructure (servers, storage memory, computing capacity, platform, etc.); nor is it necessary to install software from physical media to the computer. Instead, the software can be accessed as a service using any computer or other device (such as a mobile phone or tablet) connected to the Internet. This differentiates servitization from "software services," in which only some aspects of the software products are offered as a service (Suarez et al., 2013). Cloud computing-based servitization can be roughly divided into three service layers (Sultan, 2014a; Wortmann et al., 2012). These consist of (i) Infrastructure as a Service (IaaS), which provides computation and storage capacity, (ii) Platform as a Service (PaaS), which provides software development tools plus an application execution environment, and (iii) Software as a Service (SaaS), which provides applications on top of PaaS and IaaS (Armbrust et al., 2010; Hugos and Hulitzky, 2011, Sultan, 2014a; Wortmann et al., 2012).

Because SaaS presents a new delivery model for software (Obal, 2013; Ojala, 2016; Sultan, 2014a), there have been an increasing number of studies focusing on SaaS providers' strategies in the market. For instance, Susarla et al. (2009) studied how two forms of ex-post transaction costs[5] (monitoring costs and maladaptation costs) impact on

[5] According to Ang and Straub (1998, p. 549), transaction costs refer to "the effort, time, and costs incurred in searching, creating, negotiating, monitoring, and enforcing a service contract between buyers and suppliers."

contracts between SaaS providers and customers. It appeared that transaction costs tended to lead to information asymmetry, possibly resulting in contractual incompleteness and opportunism. In addition, it was found that for the service provider a cost-based rent fee becomes more attractive than a fixed rent fee if there is uncertainty concerning the specification of service requirements, or if there is interdependence between the SaaS application and the IT system on the customer's side. However, for the purposes of cost reduction, a fixed rent fee is more attractive for customers (Susarla et al., 2009). In a later study, Susarla et al. (2010) studied challenges in service disaggregation between the SaaS provider and the customer. They suggested that there are knowledge interdependencies between SaaS providers and their customers, and that these create challenges in combining IT services.

Several studies have focused on the adoption of SaaS (e.g. Benlian and Hess, 2011; Hsu et al., 2014; Wu et al., 2011). From the findings of Wu et al. (2011), it appears that the adoption of SaaS is more related to strategic benefits than to economic benefits. In addition, SaaS adoption is more related to subjective risks than to technical risks. Benlian and Hess (2011) focused on IT executives' perceptions of the opportunities and risks in SaaS adoption. They found that security threats were the most dominating factor in risk perception, whereas opportunities related to the cost advantages of SaaS were the strongest driver for SaaS adoption. Fan et al. (2009) studied short-term and long-term competition between SaaS providers and traditional software providers. According to their findings, SaaS can reduce price competition and facilitate differentiation from traditional software providers. However, SaaS providers may incur significant operating costs due to the fact that they have to invest in service and system capacity in order to guarantee availability of the service (Fan et al., 2009).

2.2. Software renting and pricing

As discussed above, servitization of the software offering into an SaaS model is changing traditional ways of doing business. The SaaS model facilitates delivery of software and enables software renting with a variety of pricing models. To better understand the benefits of renting, economists have compared the advantages of renting with the

advantages of buying a product outright (Bucovetsky and Chilton, 1986; Bulow, 1982; Flath, 1980). According to the literature (Flath, 1980, p. 247), renting can be defined as "a contractual arrangement for trading the rights to temporary use of an object, but not the right to all possible future use." Thus, in a rental agreement, a customer does not get the full ownership rights over the object rented, as distinct from ownership following purchase. However, there are always trade-offs between the benefits of full ownership and those of "partial ownership" – i.e. renting.

From the customer's point of view, these benefits are related to the characteristics of the product and the time period needed for usage of the product (Bulow, 1982; Flath, 1980; Tang and Deo, 2008). In other words "the shorter is one's expected tenure of use of a good, the greater are the transacting cost gains to his leasing it rather than purchasing it outright" (Flath, 1980, p. 249). According to Choudhary et al. (1998) the reasons why a customer may rent software in preference to buying it are as follows: (i) the software is for use in a short-term project, (ii) the customer may simply want to gain experience of using the software, (iii) the customer wants to test and evaluate the usability of the software, or (iv) the customer wants to avoid negative network externality. Software renting may also benefit both the software vendor and the customer by providing cost savings for customers, with higher profits also for software vendors (Choudhary et al., 1998). Software renting in the SaaS model also lessens the customers' need to have their own IT personnel and IT infrastructure. This decreases the total cost of ownership and reduces the hidden costs that may increase a firm's IT spending by as much as 80 percent in the case of traditional software licensing (Waters, 2005).

From the software provider's perspective, renting decreases transaction costs related to identifying, assuring, and maintaining quality (Choudhary, 2007; Choudhary et al., 1998; Flath, 1980; Varian, 2000). Renting can also increase the positive network externality effect (Liebowitz and Margolis, 1994), owing to the lower initial costs for customers compared to purchasing. The low costs increase the number of customers, and consequently increase the information available in the market regarding the product. Overall, this decreases customers' search costs (Choudhary et al., 1998; Porter, 2001). The benefits of renting become more complex when the rented product is in the category of information goods, as in the case of music, videos, books, journals, or computer

software (Choudhary, 2007; Sundararajan, 2004; Varian, 2000). For instance, unlike the situation with other durable goods, the ownership of the software belongs to the producer, whether the software is rented or not (Choudhary et al., 1998), and the impact of psychological ownership may be less significant (cf. Durgee and O'Connor, 1995).

When software is rented via the SaaS model, it makes possible a variety of interesting pricing models for software firms (Baranwal and Vidyarthi, 2015; Li et al., 2016; Laatikainen et al., 2013; Ojala, 2013). Hence, the actual rental fee might be based on (i) assessment base pricing, (ii) price discrimination, and (iii) price bundling, or combinations of these. In assessment base pricing, a firm charges customers using *usage-dependent* pricing, *user-based* pricing, or a combination of the two (Li et al., 2016; Lehmann and Buxmann, 2009). In the usage-dependent model, the rental fee is related to the number of transactions, the memory requirements, the duration of the running of the software, and so on. In the user-based pricing model, customers pay a fixed rent fee for unlimited use – use being based on named users, concurrent users, etc. (for a more comprehensive review, see Lehmann and Buxmann, 2009).

Price discrimination refers to a pricing model in which software is rented to different customers at different prices (Hinz et al., 2011; Lehmann and Buxmann, 2009). The price discrimination model can be divided into first-degree, second-degree, and third-degree price discrimination. In first-degree price discrimination, the software rental fee is set according to the customer's ability/willingness to pay (Hinz et al., 2011). Second-degree price discrimination refers to the quantity, time (season or duration), version, or market segmentation that is used to set the rental fee (cf. Lehmann and Buxmann, 2009; Linde, 2009). In third-degree price discrimination, customers are charged according to identifiable sectors, such as occupation, location, or age (Adachi, 2005; Laatikainen et al., 2013). In addition, software firms may combine price discrimination models (cf. Lehmann and Buxmann, 2009), applying different degrees or proportions of each.

Price bundling refers to a pricing model that includes several components, for example software products and/or services packaged together and rented to customers for a fixed price (Hui et al., 2012; Lehmann and Buxmann, 2009; Linde, 2009). By bundling different products and/or services together, software providers can make their offering

more attractive to customers (Hui et al., 2012; Linde, 2009) than by offering the same products separately. Price bundling can be divided into *offer, product, degree of integration,* and *price level* aspects (see Lehmann and Buxmann, 2009, for a more detailed review).

2.3. Competitive strategy

Servitization has posed challenges to existing business strategies (Bustinza et al., 2015). Because servitization makes possible new ways to offer, deliver, and price software, it has a direct impact on competition in the software industry, on the development processes of software (Durkee, 2010), and on the competitive strategies that software firms apply in the market. In general, a competitive strategy[6] can be seen as including three generic strategies that a firm might pursue, either as a single strategy or in combination (Porter, 1980). The first strategy, *cost leadership,* is based on price competition in situations where a firm can outperform the competition by providing a corresponding product or services at a lower price, and thus increasing the market share. The second strategy, *differentiation,* is based on the notion that a firm may develop something unique that differentiates it from competitors. The third strategy, *focus,* refers to the notion that a firm can start to focus its offering by servicing a particular target group extremely well. Porter (1979, 1980, 2008) defines five competitive forces that shape these strategies. These are (i) the threat of new entrants, (ii) the bargaining power of buyers, (iii) the bargaining power of suppliers, (iv) the threat of substitute products or services, and (v) rivalry among existing competitors.

Software markets are relatively easy to enter (Giarratana, 2004), and this increases the threat of new entrants. New entrants in the industry put pressure on prices and decrease the profitability of the industry (Porter, 2008). To avoid competition with new entrants, software firms have to differentiate their offering from other products in the market. According to Oza et al. (2010), this requires ongoing improvements to the product, aimed at enhancing the user experience, and satisfaction with the software.

[6] In accordance with Porter (1996, p. 68, 70, 75), strategy involves "the creation of a unique and valuable position, involving a different set of activities... …making trade-offs in competing... [and] …creating fit among a company's activities."

The bargaining power of buyers is high if there are only few buyers, if products are standardized, or if customers have low switching costs (Porter, 2008). Thus, if a software vendor is offering standardized products with a low switching cost, customers have more power to force prices down. In contrast, customers using enterprise software may face very high switching costs, due to the fact that changing from one form of enterprise software to another requires a large number of labor-intensive projects, including data migration and configuration (Brydon and Vining, 2008; Chen and Hitt, 2006; Porter, 2008). However, standardization involving well-executed improvements to the product (Oza et al., 2010) may make the product attractive for new customers.

The bargaining power of suppliers is high in cases where suppliers can offer differentiated products, where there are no substitutes in the market, or where buyers face high switching costs if they change the supplier. Powerful suppliers can charge higher prices or shift the costs to the buyers (Porter, 1980, 2008). Porter (2008) uses Microsoft as an example of a powerful supplier in the computer hardware industry, in which hardware manufacturers are highly dependent on Microsoft's pricing strategies. SaaS providers are also extremely dependent on their suppliers. In many cases, they need access to IaaS and PaaS providers in order to make their services available to their customers. There may be only a limited number of IaaS and PaaS providers available, increasing the bargaining power of these providers over the SaaS firms. However, an SaaS provider who can develop good contacts with these suppliers increases its ability to deliver a reliable SaaS offering.

The threat of substitute products or services may decrease the profitability of an industry (Porter, 2008). A substitute is a product or service that "performs the same or a similar function as an industry's product by a different means" (Porter, 2008, p. 84). The threat of a substitute is high if a new product offers better value at a more attractive price than the older one. In cloud computing, new services such as video-on-demand can be seen as a substitute for traditional video film renting; such a substitute will decrease the profitability of video rental outlets (Porter, 2008).

Rivalry among existing competitors can impact on prices. According to Porter (2008), rivalry may decrease prices, for example (i) if the products are similar and there are low switching costs for customers, or (ii) if the product is perishable. As discussed

above, switching costs in software may be high in the case of enterprise software (e.g. enterprise resource planning (ERP) software) and lower in that of more standardized consumer software (as in the case of word processing software). Perishability may also impact more on standardized software products. Thus, Porter (2008) uses computers as an example of products that become outdated fairly quickly. The same can be seen in the software industry, where there is an ongoing need for new software versions or updates, especially in the case of standardized software products (Nambisan, 2013; Suarez et al., 2013).

2.4. Synthesis

The servitization of the IT industry and the development of cloud computing have brought new opportunities for software firms to sell and deliver their products using the SaaS model (Armbrust et al., 2010; Hugos and Hulitzky, 2011; Iyer and Henderson, 2010). SaaS brings new possibilities to compete in the market (Ojala, 2016) and to create revenue and pricing models that are attractive to customers. Economic theories concerned with renting, and subsequent studies focusing on software renting and pricing, have increased our general understanding of revenue and pricing models in a monopolistic market situation. However, most of the studies have used analytical approaches to reveal the core parameters of entrepreneurial decision-making, or have used binary variables, without focusing on possible mixed revenue models or hybrid pricing mechanisms in a situation of market competition. Thus, there remains a need for a more robust theoretical understanding of how market competition impacts on software revenue and pricing models.

3. Methodology

The research method selected for this study covered a real-life environment in which there was a decision-making process related to revenue models. It was important that the method should cover human actions, enable an in-depth investigation of the complex phenomena at work, and capture cause-and-effect relationships. With all this in mind, the

present study applied a multiple case study methodology similar to the approaches presented by Eisenhardt (1989) and Yin (2009).

The research setting for this study consisted of five software firms that acted as SaaS providers. Since the sample used will necessarily influence the results of the study (Miles and Huberman, 1994), multiple criteria were used to select the cases, following the recommendations by Eisenhardt (1989), and by Eisenhardt and Graebner (2007). In terms of theoretical considerations, the following aspects were seen as relevant: (i) the case firms were developing their software for different industries, (ii) the sample included both relatively old firms and recently established firms, and (iii) three of the firms also made traditional software licenses available for their software at the time of the interviews. Another highly important criterion was good access to the required information. In the present instance, three of the firms were involved in a national cloud software project in Finland, while two of the firms could be contacted on the basis of the author's knowledge of the software industry. These aspects increased mutual trust between the researcher and the case firms, and facilitated the collection of relevant information, as recommended by Stake (1995). Altogether, the case firms represented a wide variety of SaaS early adopters, and furthermore, could be seen as critical cases that might serve as examples for other firms considering various revenue and pricing models. It should be noted that this kind of coverage is important for studies when one has only a small sample of firms (Eisenhardt, 1989; Pettigrew, 1990).

Multiple sources of information were used to gather data on each case firm. The main form of data collection consisted of in-depth interviews. Altogether, 5–10 interviews per firm were conducted, each lasting 45–90 minutes. Thus, altogether 37 semi-structured open-ended interviews were carried out for this study. The interviewees consisted of chief executive officers (CEOs), sales managers, vice presidents, members of the board of directors, and software engineers (see Table 1). These persons were selected because of their knowledge of the firm's revenue and pricing strategies. The interviews with the CEOs were the main source of information, and the CEOs provided help in identifying relevant interviewees from the case firms.

During the first interview, general information on the firm was collected, including its history, products, customers, partners, and so on, in addition to discussion of

the actual revenue model. An open-ended interview structure was adhered to, using themes from previous literature. The first interview with the firm lasted approximately 90 minutes. In the second and following interviews, more structured interview guidelines were used, based on the information gathered in the previous interview(s). These subsequent interviews focused on the revenue models in detail; they were tailored to the interviewee's role in the firm and to his/her involvement in pricing strategy. The author recorded all the interviews, and personally transcribed them verbatim.

Table 1. Overview of the data sources.

Firm	Informant title	Number of interviews with a given informant	Total number of interviews per case firm
Firm A	Co-founder/CEO	1	8
	Vice President (General management)	4	
	Vice President (Sales)	3	
Firm B	Co-founder/CEO	6	10
	Chairman, Board of Directors	1	
	Vice President (Software Engineering)	1	
	Executive Director (Corporate Planning)	1	
	General Manager (Global Management)	1	
Firm C	CEO	3	5
	Sales Manager	2	
Firm D	Co-founder/CEO	2	6
	Strategic Accounts Manager	2	
	Sales Manager	2	
Firm E	Co-founder/CEO	2	8
	Co-founder/Art Director	2	
	Co-founder/CTO	1	
	COO/Chairman of the Board	1	
	Head of Sales	1	
	Sales Engineer	1	

The method utilized in the data analysis was content analysis. The analysis of the case data consisted of three concurrent flows of activity (Miles and Huberman, 1994): (i) data reduction, (ii) data displays, (iii) conclusion-drawing/verification. In (i) the data reduction phase, the data were given focus and simplified through compilation of a detailed case history of each firm. This is in line with Pettigrew (1990), who suggests that organizing incoherent aspects in chronological order is an important step in understanding the causal links between events. On the basis of the interviews and other material collected from the case firms, tables were used to identify and categorize the unique patterns of each case under sub-topics derived from the research question. These sub-topics included the

following categories: (i) the firm's product offering and its customers, (ii) revenue model(s) and reasons for usage of the revenue model(s), (iii) the pricing model and reasons for usage of the pricing model(s), and (iv) the competitive advantages of the revenue model. These data were used to write the case descriptions of each firm, and the summaries of the case descriptions are presented in Table 2. In addition, checklists and event listings were used to identify factors impacting on the selected revenue model (Miles and Huberman, 1994).

In the data display phase, the data from the case descriptions were arranged in new tables and figures. The tables included direct citations from the interview data, and figures were used to illustrate the findings graphically (Miles and Huberman, 1994). In the phase of conclusion-drawing and verification, the cross-case aspects that appeared to have significance for this study were identified. At this stage regularities, patterns, explanations, and causalities related to the phenomena were noted. This was helpful in keeping the empirical findings and theory at the forefront of the article (see Eisenhardt and Graebner, 2007), and in ensuring that the findings corresponded to the research question set in the introduction.

To ensure the validity of the data collected, the following steps were taken. Firstly, at least two employees from a case firm were interviewed, to avoid bias from individual opinions (Huber and Power, 1985; Myers and Newman, 2007; Eisenhardt and Graebner, 2007). Secondly, the complete transcripts were sent back to the interviewees for review, to ensure the correctness of the transcripts and avoid misunderstandings. For the most part, the interviewees accepted the transcript in the form in which it was sent to them. However, in some cases, the interviewees gave some minor comments related to the misspelling of a partner's name or to some particular wording. Thirdly, different kinds of secondary data[7] were used to validate and "triangulate" (Miles and Huberman, 1994) the data collected from interviews. Fourthly, if there were inconsistencies within the interviews or between interviews and the secondary data, these were discussed with the interviewee and the CEO to avoid misunderstandings and retrospective bias (Huber and Power, 1985). Finally, to ensure the accuracy of the categorization of the data into

[7] The secondary data included internal and external memos, promotion material, press releases, websites, and brochures.

tables and figures, an external researcher was consulted in order to validate the categorization. The author provided him with access to the case transcripts, and he reviewed the correctness of the categorization. Thereafter, we discussed the categorization of the factors; two items were recategorized on the basis of the discussion.

Table 2. Overview of the case firms and factors impacting on the selection of a revenue model.

----- Insert Table 2 around here ----

4. Findings

This section presents the findings of the qualitative case study. Table 2 gives an overview of the factors impacting on the choice between software renting and software licensing. In order to give a clear description of (i) the factors that drove case firms to *rent* their software, and (ii) the factors that drove case firms to *license* their software, the findings are presented as a cross-case study rather than describing each individual case separately.

4.1. Factors driving the case firms to rent their software

The case firms A, C, and D used both software renting and software licensing as revenue models, whereas firms B and E currently used only software renting[8]. The firms favored software renting as the first option because (i) it decreased development costs[9], (ii) it helped them to expand the customer base, (iii) it helped in differentiating them from competitors, and (iv) it allowed flexible pricing. In addition, all the case firms with a

[8] It should be noted here that all three revenue models, i.e. (i) software licensing, (ii) software renting, and (iii) pay-per-use, were available to the firms. However, they applied solely software renting or a combination of software renting and software licensing at the time of interviews.

[9] As distinct from transaction costs, development costs include "the physical or other primary processes necessary to create and distribute the goods or services being produced" (Wigand and Benjamin, 1995).

license model announced that they were moving increasingly toward servitization of their offering, and a software-renting model. These points will be considered individually, below.

(i) The rental model through servitization decreased the development costs of case firms A and D (i.e. costs related to the installation, delivery, implementation, maintenance, and after-sales support of the software). Thus, the case firms knew that their customers were using the same version of the software, and by means of the public cloud, the case firms were able to bring updates that were visible to all their customers immediately. The technical features of the model consistently brought cost savings to the case firms and made it possible to offer the software at a lower price. By means of this strategy the firms could improve their cost leadership in the market and protect their business against rivalry and substitutes. The CEO of Firm A explained this as follows:

"It brings cost savings. If we sell the Intranet version, it has to be installed in the customer's premises, so it requires far more resources from us…in many cases, we need an employee who will go and meet the customer, install the software, implement the software, and give support. And then all the updates have to be delivered separately to each customer. In the cloud model, all this is centralized."

(ii) Software renting made it easier to expand the customer base, especially from large-sized customers toward small and medium-sized customers as was done by case firms C and E. In other words, servitization of the offering and software renting helped the firms to expand their customer base, since the rental solution made the service attractive also for smaller customers, who might not have a budget for the initial investments required by traditional software licensing. Hence, software renting shifted customers' capital investment onto operational costs: smaller customers could then start to use the software without special budgeting, or without having to obtain the approval of top management. This also helped vendors to protect their business against rivalry and substitutes, and gave cost leadership advantages. The CEO of Firm C commented on this as follows:

"Previously we only had a traditional licensing model – an initial license fee plus an annual maintenance fee. However, we are now increasingly moving towards a model in which we charge a monthly rental fee. Then customers don't have to make an investment decision – the customer just pays the monthly rental fee. Then it is more like a cost, not an investment."

(iii) Software renting through servitization helped the case firms (A, B, D, and E) to differentiate themselves from competitors who used traditional licensing and packaged software solutions. It also allowed faster and more cost-effective delivery, maintenance, and after-sales services. Thus, the firms were able to use differentiation as a competitive advantage, offering benefits that their competitors did not have. In this way the case firms acted against rivalry in the software market. The CEO of Firm E explained the competitive advantage of servitization in the following manner:

"The most notable difference between us and our competitors is that they use CAD-based programs that have to be licensed for a workstation… we have changed the way of delivering this as a service… the product is available over the Internet connection and you can learn how to use it in one minute, whereas for a CAD-based program, you have to take a course before you know how to use it."

(iv) Software renting enabled flexible pricing based on the number of users, the functionalities used, and so on, in a way similar to licensing options. In addition, rental made it possible to change the pricing according to the named users or concurrent users, during or after the rental period (based on the rental agreement). Furthermore, it made it possible to offer the software at reasonable price if a customer needed it only for short-term usage. This all provided cost leadership advantages for the case firms. In firms A, C, D, and E the pricing was based on concurrent user assessment, taking into account the need to protect their business against increasing maintenance costs, whereas Firm B used named user assessment and price bundling. Thus, the number of concurrent users increased the price, since the number of users correlated with the capacity and costs required for data storage and computing power. A larger number of users also made

installation and customer support more complex and time consuming. In addition to concurrent user-based pricing, Firm A and Firm B used time-based price discrimination to protect their business against low switching costs. Hence, the fee for a short-term rental agreement was higher than for a long-term contract. This helped the firms to avoid rivalry or the threat of substitute products in the market. The vice president of Firm A commented on this as follows:

"Of course we can rent the software for a shorter time if a customer needs it, for example for a short project – that is possible. However, it might become more expensive for the customer since the initial costs are the same for us, regardless of the rental time. But if there is a need for it, we are ready to discuss it with the customer."

In contrast, firms D and E protected their business against rivalry and substitutes by using separate pricing for the implementation work. The implementation of their software required so much labor-intensive project work that customers' switching costs increased, offsetting the benefits of short-term renting. In firms C and E, the rental fee was also based on version-based price discrimination. In addition, Firm E used quantity-based price discrimination, dependent on the number of elements that a customer wished to include in the software. This means that in the servitized offering, the functionality-based pricing used in software licensing was transformed into a rental model.

In the case of Firm B, the bargaining power of the supplier impacted on the revenue model selected. The firm used network operators as an IaaS provider and delivery channel to end-users. For the network and telecom operators it was normal to use renting as a revenue model for their service offerings, and because operators have so much bargaining power over the content providers, there were no other options, in practice, that Firm B could use. However, they were still able to use a variety of pricing criteria for their games. As the CEO of Firm B put it:

"In practice, we could also use a pay-per-use model, for instance if a player plays the game let's say fifteen minutes, we will charge him a price unit that corresponds to fifteen minutes. However, network operators have become accustomed to using rental models

and that is why we have to use one too. It is not always a model that the game developers [the content providers] would like us to use, or that we would want to offer, but we are negotiating between two big players [network operators and game developers]… and the operators have a strong role, as they deliver the service to the players."

4.2. Factors inhibiting the case firms from renting their software

The competitive forces that drove case firms A, C, and D to license their software to some of the customers instead of renting were related to the bargaining power of the buyers. Certain customers were so important that the software firms were willing to offer their product in the form of a license. The reasons for customers to prefer a license were related to: (i) security concerns, (ii) a low readiness for new technologies, and (iii) the IT infrastructure available to them. These aspects will be considered in more detail below.

Considering the first point, some of the customers of firms A, C, and D favored software licensing plus usage of the software in their internal data center, because of security concerns over highly sensitive data. Following this route increased the feeling of psychological ownership and made customers less dependent on the software provider. Firm D had also found that the IT policies of some large-sized customers specifically restricted software renting and/or usage of the software through the public cloud. These polices were strictly rooted in the customers' organizational culture and could not easily be changed. This meant that the only option was to sell a traditional license for the customer's internal data center. In contrast, firms B and E currently used only software renting, as they did not experience problems with security concerns. This was mainly because of the software offerings (video games and 3D modeling software), which were aimed at activities in which there were no security issues for the customer. The CEO of Firm E explained this in the following manner:

"There are no data protection threats in our product [3D modeling software]. It is a sales tool for marketing, and its purpose is to sell the product that we have modeled. It is public marketing information. Maybe some customers don't want us to model a product

in too much detail. However, anyone can go to the store and see the real, physical product there, and how it is constructed."

In relation to the second point, Firm A reported that in some cases, customers had a low readiness to adopt new technologies, and this affected the revenue model in use. The conservatism of the customers favored software licensing, even if software renting would clearly have been a more cost-effective way to purchase the software.

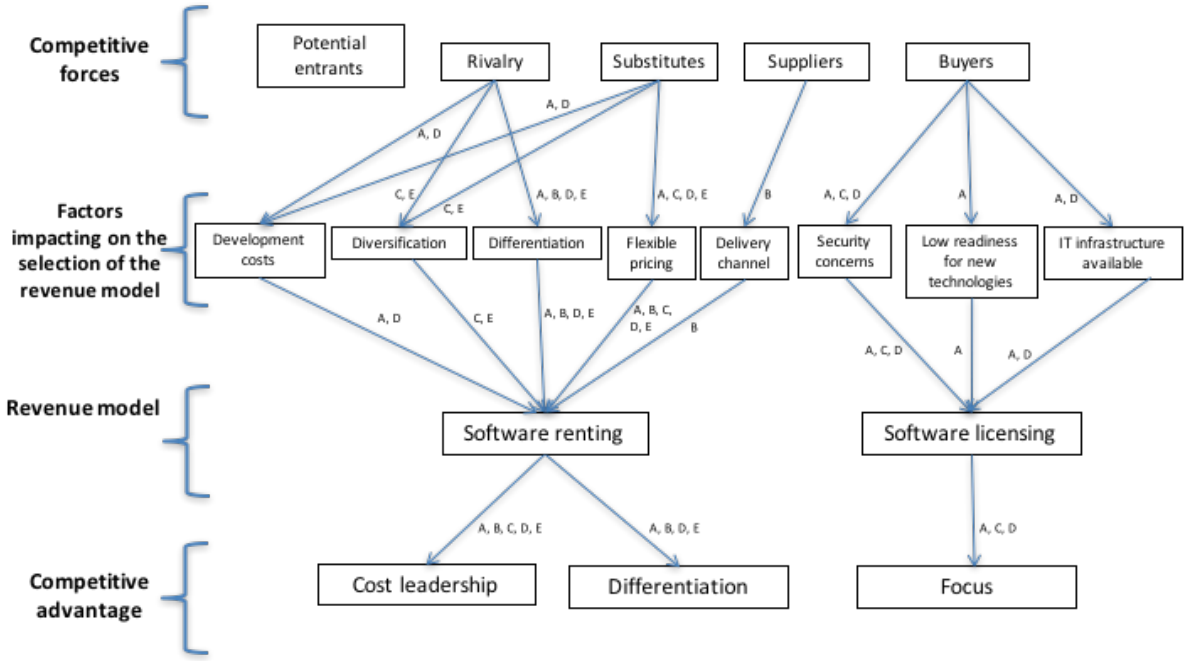
As for the third point above, firms A and D had large customers who possessed their own data centers, with IT personnel on hand. Customers in this category were more willing to buy a traditional license and to use the software within their own premises. In contrast, Firm E had many small customers who were connected to the Internet but did not have in-house server facilities to run an Intranet. In such cases, renting through servitization was a cost-effective way to get access to the necessary software, storage space, and computing capacity. Altogether, the findings demonstrate how firms A, C, and D used a focus strategy to offer software via a software license for a particular customer group.

5. Discussion

Figure 1 summarizes the main findings of the study. It indicates how competitive forces can be linked to the choice of revenue model. Depending on the competitive situation in the market, lower development costs, diversification, differentiation, and/or flexible pricing in software renting seem to help software vendors to protect their business against rivalry and substitutes (cf. Porter, 1980, 2008). This is mainly because of the lower initial costs and the more differentiated offering for customers as compared to software licensing. Software renting can help to achieve differentiation from competitors who use traditional licensing and packaged software solutions. The findings also show how a powerful supplier may restrict the revenue models to be applied; thus, a powerful IaaS provider was able to limit the revenue model solely to software renting. In addition, the bargaining power of buyers led some software providers to use software licensing in

addition to software renting. Altogether, one can see how software renting was related to cost leadership and/or differentiation strategies, whereas software licensing was connected to a focus strategy.

Figure 1. Impact of competitive forces on the choice of revenue model. The letters correspond to the case firms.



The findings here complement and expand on earlier studies on software renting and related pricing models. It appears that the selection of the revenue model does not always involve an exclusive choice between software renting and licensing, since the firms can use these models in parallel depending on the competitive forces (Porter, 1980) in the market. In addition, a great variety of software pricing models are available, and firms can use these individually or in combination to compete in the market. These findings expand from earlier economic literature (Bucovetsky and Chilton, 1986; Bulow, 1982; Flath, 1980), and studies on software renting (Choudhary et al., 1998; Choudhary, 2007), which have tended to use predetermined variables applied to monopolistic market situations, and to ignore possible mixed revenue models or hybrid pricing mechanisms in situations of market competition. In addition to the selection of the revenue model, the

selection of a proper pricing model is not always a simple choice between binary models, as posited by Sundararajan (2004) and Susarla et al., (2009, 2010); in fact, the selection of the most appropriate pricing model will depend on the competitive forces in the market.

In line with previous studies (Lehmann and Buxmann, 2009; Susarla et al., 2009), the findings here suggest that software providers tend to favor fixed concurrent or named user-based pricing models. This is counter to the widespread assumption that cloud-based delivery is usage dependent (see e.g. Armbrust et al., 2010; Dikaiakos et al., 2009; Louridas, 2010). However, the findings also demonstrate how firms can use price discrimination and price bundling, in addition to user-based pricing, in order to define the final price, taking into account competition in the market. Hence, software providers may seek to protect their business against low switching costs by using second-degree price discrimination based on time criteria. This finding complements those of Oza et al. (2010) in suggesting that in addition to improvements in user experience and security, SaaS providers can protect their business against low switching costs by using an appropriate pricing model. However, in the case of enterprise software, the implementation of the software may require so much labor-intensive project work that the switching costs increase, offsetting the benefits of short-term renting for the customer.

Interestingly, previous studies (Choudhary et al., 1998; Susarla et al., 2009; Varian, 2000) have indicated that transaction costs are lower in software renting. However, none of the case firms mentioned this as an advantage. This may be due to the fact that in software renting, the negotiation costs and contract monitoring costs related to renting offset the benefits of other transaction cost advantages (advantages which would include the lower costs of identifying, assuring, and maintaining quality; see Choudhary et al. (1998) and Varian (2000)). Nevertheless, the present study does indicate that the rental model decreases development costs and consequently makes software renting cost-effective and attractive for both the software vendor and the customer. Thus, the findings here go beyond existing studies insofar as they show how lower development costs can be used as a competitive strategy. The lower development costs of software renting were valuable tools against rivalry and substitute products, since they offered similar or better performance, with lower costs, than other products in the market (cf. Porter, 2008).

Software renting also helped to diversify the customer base by making software products available to smaller customers. The factor of lower development costs gives empirical support to a model encompassing competition between SaaS providers and traditional software providers (Fan et al., 2009). However, differentiation was based not only on lower implementation costs – as argued by Fan et al. (2009) – but also on faster and more cost-effective delivery, maintenance, and after-sales services within the SaaS model.

Data security concerns played a significant role in the adoption of SaaS – an aspect examined by Benlian and Hess (2011), Hsu et al. (2014), and Wu et al. (2011). The findings reported here expand on these studies by demonstrating how data security concerns are related to the revenue model of a firm, and how they impact on competitive forces in the software market. The findings also demonstrate how SaaS providers may be forced to offer their product under license for customers with high bargaining power. Otherwise, the firm will simply lose these customers. This notion also gives empirical support to the view expressed by Choudhary (2007) to the effect that SaaS impacts on the relative bargaining power between buyers and sellers.

In the present study, the customer's IT knowledge appeared to have a strong impact on deciding whether to buy or to rent the software. If customers had already invested in IT infrastructure and their own IT personnel, they saw traditional software licensing as a more attractive choice. This was mainly because the benefits of software renting became less significant, and operating the software in-house increased the feeling of trust. This is an aspect neglected in previous studies on software renting (Choudhary et al., 1998; Choudhary, 2007). However, the findings hare take the findings of Obal (2013) a stage further, since they demonstrate how interorganizational trust can impact on the decision to either license or rent the software.

6. Conclusions

As a theoretical contribution, this study builds on Porter's (1980, 2008) theory of competitive strategy by demonstrating the interaction between competitive forces and revenue models. First of all, the study shows how software renting may affect the competitive situation in the market by protecting software providers against rivalry and

substitutes. In addition, competitive forces, such as the bargaining power of buyers and suppliers, influence the software vendor's selection of revenue model. Secondly, the study contributes to the theory of competitive strategy (Porter, 1980, 2008) in the context of IT servitization (Bustinza et al., 2015), in indicating that servitization makes it possible to use different competitive strategies flexibly, according to market competition and customers' preferences. This means that the selection of the competitive strategy is not always a straightforward choice between a cost leadership, differentiation, or focus strategy. Thirdly, the findings here build on earlier works on software renting and pricing. In a situation where servitization is moving revenue models from selling products towards rental contracts (Barnett et al., 2013), this study shows how software firms can apply mixed revenue models or hybrid pricing mechanisms on the basis of market competition. Finally, the findings expand from previous economic literature (Bucovetsky and Chilton, 1986; Bulow, 1982; Flath, 1980) and studies on software renting (Choudhary et al., 1998; Choudhary, 2007), which have tended to apply predetermined variables and to consider monopolistic market situations.

In addition to revealing factors relevant to the selection of the revenue model, the findings demonstrate that the selection of a proper pricing model is not always a simple choice between binary models, in the manner posited by Sundararajan (2004) and Susarla et al. (2009, 2010); also that the selection of the most appropriate pricing model will depend on the competitive forces (Porter, 1980) in the market.

From a managerial perspective, the servitization of software offerings should allow software vendors to expand their business opportunities considerably. Several factors will interplay in this. One point to note is that although servitization is a highly promising way of distinguishing oneself from the competition in the software industry, the entry barriers in the cloud business are relatively low. This means that in the future there may well be an increase in new entries when software firms servitize their traditional software offering – similar to the situation that occurred in e-commerce during the IT boom (Porter, 2001). Thus, software vendors will be obliged to protect their business against new entries and, in addition, to have attractive pricing, if they are to respond to the constantly changing needs of the market. This will require flexibility and ongoing recognition of new business opportunities, given the unpredictabilities in the

development of the software market, and the competition within it. A further point to note is that changes in software architecture are commonly much harder to implement than changes in revenue models, and that it is challenging or even impossible to move traditionally developed "packaged" software to the cloud environment. Nevertheless, all the case firms had planned their software architecture to be compatible with both the cloud environment and stand-alone implementations. This is not the case with most incumbent software vendors, whose software cannot be readily moved to the cloud environment. Such vendors must either rewrite their software or alternatively, focus their development efforts on new product lines whose architecture is already compatible with the cloud, while continuing to maintain their traditional software until it is phased out.

This study also points to aspects requiring further research. One noteworthy feature is that when a firm does not provide the software directly to the customer as an on-premise solution, it requires intensive cooperation with other support service providers such as PaaS and IaaS providers if it is to acquire the resources needed. There is a need for research on how relationships with these firms can be established, and on how the formation of these relationships impacts on the profitability of given firms. It is also worth noting that all the case firms in this study were relatively small. Although small software firms can be very successful (Parker et al., 2016) the findings might not be fully generalizable to large software firms. Furthermore, the findings here need further validation via quantitative methods, bearing in mind that the aim here was primarily to understand the phenomena and the cause-effect relationships involved. Hence, one must avoid generalizing too broadly from such a limited set of data. A final point to note is that this study does not take into consideration the actual *success* of the case firms' revenue and pricing models. Thus, it would be of interest to conduct quantitative studies in order to estimate how successful particular revenue and pricing models turn out to be.

Acknowledgements

I wish to thank Pasi Tyrväinen and the four anonymous reviewers for their valuable comments and suggestions. In addition, I wish to thank Markus Salo for his assistance in

data validation. The research has been generously funded by the Foundation for Economic Education (Liikesivistysrahasto) and by the Marcus Wallenberg Foundation.

References

Adachi, T. (2005) Third-Degree Price Discrimination, Consumption Externalities and Social Welfare. Economica 72(285): 171-178.

Ahonen, J., Savolainen, P., Merikoski, H. & Nevalainen, J. (2015) Reported project management effort, project size, and contract type. Journal of Systems and Software 109: 205-213.

Ang, S. & Straub, D.W. (1998) Production and Transaction Economies and IS Outsourcing: A Study of the U.S. Banking Industry. MIS Quarterly 22(4): 535-552.

Armbrust et al. (2010) A view of cloud computing. Communication of the ACM 53(4): 50-58.

Baines, T.S., Lightfoot, H.W., Benedettini, O. & Kay, J.M. (2009) The servitization of manufacturing: A review of literature and reflection on future challenges. Journal of Manufacturing Technology Management 20(5): 547-567.

Baranwal, G. & Vidyarthi, D.P. (2015) A fair multi-attribute combinatorial double auction model for resource allocation in cloud computing. Journal of Systems and Software 108: 60-76.

Barnett, N., Parry, G., Saad, M., Newnes, S.B. & Goh, Y.M. (2013) Servitization: Is a Paradigm Shift in the Business Model and Service Enterprise Required? Strategic Change 22(3-4): 145-156.

Bass, L., Clements, P. & Kazman, R. (1998) Software Architecture in Practice. SEI series in software engineering. Addison Wesley Longman, Inc. USA.

Benlian, A. & Hess, T. (2011) Opportunities and risks of software-as-a-service: Findings from a survey of IT executives. Decision Support Systems 53(1): 232-246.

Brydon, M. & Vining, A.R. (2008) Adoption, Improvement, and Disruption: Predicting the Impact of Open Source Applications in Enterprise Software Markets. Journal of Database Management 19(2): 73-94.

Bucovetsky, S. & Chilton, J. (1986) Concurrent renting and selling in a durable-goods monopoly under threat of entry. RAND Journal of Economics 17(2): 261-275.

Bulow, J.I. (1982) Durable-Goods Monopolists. Journal of Political Economy 90(2): 314-332.

Bustinza, O.F., Ziaee, B.A., Baines, T.S. & Elliot, C. (2015) Servitization and competitive advantage: the importance of organizational structure and value chain position. Research Technology Management 58(5), 53-60.

Chen, P.Y. & Hitt, L.M. (2006) Information technology and switching costs. Handbook on Economics and Information Systems 1, 437-470.

Choudhary, V. (2007) Comparison of Software Quality Under Perpetual Licensing and Software as a Service. Journal of Management Information Systems 24(2): 141-165.

Choudhary, V. Tomak, K. & Chaturvedi, A. (1998) Economic Benefits of Renting Software. Journal of Organizational Computing and Electronic Commerce 8(4): 277-305.

Cusumano, M.A. (2007) The changing labyrinth of software pricing. Communication of the ACM. 50(7): 19-22.

Cusumano, M.A. (2010) Cloud computing and SaaS as new computing platforms. Communication of the ACM 53(4): 27-29.

Dikaiakos, M.D., Katsaros, D., Mehra, P., Pallis, G. & Vakali, A. (2009) Cloud Computing: Distributed Internet Computing for IT and Scientific Research. IEEE Internet Computing 13(5): 10-13.

Durkee, D. (2010) Why Cloud Computing Will Never Be Free. Queue 8(4): 1-10.

Durgee, J.F. & O'Connor, G.C. (1995) An Exploration into Renting as Consumption Behavior. Psychology & Marketing 12(2): 89-104.

Eisenhardt, K.M. (1989) Building theories from case study research. Academy of Management Review 14(4): 532-550.

Eisenhardt, K.M. & Graebner, M.E. (2007) Theory building from cases: Opportunities and challenges. Academy of Management Journal 50(1): 25-32.

Fan, M., Kumar, S. & Whinstor, A.B. (2009) Short-term and long-term competition between providers of shrink-wrap software and software as a service. European Journal of Operational Research 196(2): 661-671.

Ferrante, D. (2006) Software Licensing Models: What's Out There? IT Professional 8(6): 24-29.

Flath, D. (1980) The Economics of Short-term Leasing. Economic Inquiry 18(2): 247-259.

Giarratana, M.S. (2004) The birth of a new industry: entry by start-ups and the drivers of firm growth: The case of encryption software. Research Policy 33(5): 787-806.

Hinz, O., Hann, I. & Spann, M. (2011) Price discrimination in e-commerce? An examination of dynamic pricing in name-your-own price markets. MIS Quarterly 35(1): 81-98.

Hsu, P-F., Ray, S. & Li-Hsieh, Y-Y. (2014) Examining cloud computing adoption intention, pricing mechanism, and deployment model. International Journal of Information Management 34(4), 474-488.

Huber, G.P., and Power, D.J. (1985) Retrospective Reports of Strategic-level Managers: Guidelines for Increasing their Accuracy, Strategic Management Journal 6: 171-180.

Hui, W., Yoo, B., Choudhary, V. & Tam, K.Y. (2012) Sell by bundle or unit?: Pure bundling versus mixed bundling of information goods. Decision Support Systems 53(3): 517-525.

Huang, K-C. & Shen, B-J. (2015) Service deployment strategies for efficient execution of composite SaaS applications on cloud platform. Journal of Systems and Software 107: 127-141.

Hugos, M.H. & Hulitzky, D. (2011) Business in the Cloud: What Every Business Needs to Know About Cloud Computing. John Wiley & Sons, Inc.

Iyer, B., and Henderson, J.C. (2010) Preparing for the Future: Understanding the Seven Capabilities of Cloud Computing, MIS Quarterly Executive 9(2): 117-131.

Laatikainen, G. & Ojala, A. (2014) SaaS Architecture and Pricing Models. 2014 IEEE International Conference on Services Computing (SCC), 597-604.

Laatikainen, G., Ojala, A. & Mazhelis, O. (2013) Cloud Services Pricing Models. In Herzwurm, G. and Margaria, T. (Eds.) Software Business. From Physical Products to Software Services and Solutions. Lecture Notes in Business Information Processing, N. 150. Springer Berlin Heidelberg.

Leavitt, N., 2009. Is Cloud Computing Really Ready for Prime Time? Computer 42(1): 15-20.

Lee, K.C., Lee, H., Lee, N. & Lim, J. (2013) An agent-based fuzzy cognitive map approach to the strategic marketing planning for industrial firms. Industrial Marketing Management 42(2): 552-563.

Lehmann, S. & Buxmann, P. (2009) Pricing Strategies of Software Vendors. Business & Information Systems Engineering 1(6): 452-462.

Li, Z., Zhang, H., O'Brien, L., Jiang, S., Zhou, Y., Kihl, M. & Ranjan, R. (2016) Spot pricing in the Cloud ecosystem: A comparative investigation. Journal of Systems and Software 114: 1-19.

Liebowitz, S.J. & Margolis, S.E. (1994) Network Externality: An Uncommon Tragedy. The Journal of Economic Perspectives 8(2): 133-150.

Linde, F. (2009) Pricing information goods. Journal of Product & Brand Management 18(5): 379-384.

Louridas, P. (2010) Up in the Air: Moving Your Trust in Cloud Computing. IT Professional 12(5): 20-27.

Mayers, M.D. & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. Information and Organization 17(1), 2-26.

Miles, M.B. & Huberman A.M. (1994) Qualitative Data Analysis: An Expanded Sourcebook. California: Sage Publications.

McAfee, A. (2011) What Every CEO Needs to Know About The Cloud. Harvard Business Review 89(11): 124-132.

Nambisan, S. (2013) Information technology and product/service innovation: A brief assessment and some suggestions for future research. Journal of the Association for Information Systems 14(4): 215-226.

Obal, M. (2013) Why do incumbents sometimes succeed? Investigating the role of interorganizational trust on the adoption of disruptive technology. Industrial Marketing Management 42(6): 900-908.

Ojala, A. (2012) Comparison of different revenue models in SaaS. Proceedings of 5th Computer Games, Multimedia & Allied Technology Conference (CGAT 2012).

Ojala, A. (2013) Software-as-a-Service Revenue Models. IT Professional 15(3): 54-59.

Ojala, A. (2016) Discovering and creating business opportunities for cloud services. Journal of Systems and Software 113: 408-417.

Oza, N., Karppinen, K. & Savola, R. (2010) User Experience and Security in the Cloud – An Empirical Study in the Finnish Cloud Consortium. IEEE Second International Conference on Cloud Computing Technology and Science.

Parker, G.G., Alstyne, M.W.V., and Choudary, S.P. (2016) Platform Revolution: How Networked Markets Are Transforming the Economy - and How to Make Them Work for You. W.W. Norton & Company.

Pettigrew, A.M. (1990) Longitudinal Field Research on Change: Theory and Practice. Organization Science 1(3): 267-292.

Porter, M. (1979) How Competitive Forces Shape Strategy. Harvard Business Review 57(2): 137-145.

Porter, M. (1980) Competitive strategy: techniques for analyzing industries and competitors. Free Press, NY.

Porter, M. (1996) What Is Strategy? Harvard Business Review 74(6), 61-78.

Porter, M. (2001) Strategy and the Internet. Harvard Business Review 79(3): 62-78.

Porter, M. (2008) The five competitive forces that shape strategy. Harvard Business Review 86(1): 78-93.

Schroeder, A. & Kotlarsky, J. (2015) Digital resources and their role in advanced service provision:a VRIN analysis. In Baines, T. and Harrison, D.K. (Eds.) Servitization: the theory and impact. Aston University, UK, 67-74.

Smolander, K., Lassenius, C. & Rossi, M. (2016). Preface to the special section on software business. Journal of Systems and Software, 113, 407.

Stake, R.E. (1995) The art of case study research. Sage Publications.

Suarez, F.F., Cusumano, M.A. & Kahl, S.J. (2013) Services and the Business Models of Product Firms: An Empirical Analysis of the Software Industry. Management Science 59(2), 420-435.

Sultan, N. (2014a) Servitization of the IT Industry: The Cloud Phenomenon. Strategic Change 23(5-6), 375-388.

Sultan, N. (2014b) Making use of cloud computing for healthcare provision: Opportunities and challenges. International Journal of Information Management 24(2): 177-184.

Sun, W., Zhang, X., Guo, C. J., Sun, P. & Su, H. (2008) Software as a service: Configuration and customization perspectives. IEEE Congress on Services Part II, 2008. SERVICES-2, pp. 18-25.

Sundararajan, A. (2004) Nonlinear Pricing of Information Goods. Management Science 50(12): 1660-1673.

Susarla, A., Barua, A. & Whinston, A.B. (2009) A Transaction Cost Perspective of the "Software as a Service" Business Model. Journal of Management Information Systems 26(2): 205-240.

Susarla, A., Barua, A. & Whinston, A.B. (2010) Multitask Agency, Modular Architecture, and Task Disaggregation in SaaS. Journal of Management Information Systems 26(4): 87-117.

Tang, C.S. & Deo, S. (2008) Rental price and rental duration under retail competition. European Journal of Operational Research 187(3): 806-828.

Valtakoski, A. (2015) Initiation of buyer–seller relationships: The impact of intangibility, trust and mitigation strategies. Industrial Marketing Management 44: 107-118.

Vandermerwe, S. & Rada, J. (1988) Servitization of business: Adding value by adding services. European Management Journal 6(4), 314-324.

Varian, H. (2000) Buying, Sharing and Renting Information Goods. The Journal of Industrial Economics 48(4): 473-488.

Waters, B. (2005). Software as a service: A look at the customer benefits. Journal of Digital Asset Management 1, 32-39.

Wigand, R.T. & Benjamin, R.I. (1995) Electronic Commerce: Effects on Electronic Markets. Journal of Computer-Mediated Communication 1(3).

Wortmann, J.C., Don, H., Hasselman, J. & Wilbrink, A. 2012. Enterprise Information Systems as a Service: Re-engineering Enterprise Software as Product-Service System. In Advances in Production Management Systems. Value Networks: Innovation, Technologies, and Management, 496-505.

Wu, W-W., Lan, L.W. & Lee, Y-T. (2011) Exploring decisive factors affecting an organization's SaaS adoption: A case study. International Journal of Information Management 31(6): 556-563.

Yin, R.K. (2009) Case study research: Design and methods. CA: SAGE Publications.