



Queensland University of Technology
Brisbane Australia

This may be the author's version of a work that was submitted/accepted for publication in the following source:

Stuckenberg, Sebastian, [Fielt, Erwin](#), & Loser, Timm
(2011)

The impact of software-as-a-service on business models of leading software vendors: Experiences from three exploratory case studies.

In Seddon, P B & Gregor, S (Eds.) *Proceedings of the 15th Pacific Asia Conference on Information Systems (PACIS)*.

Queensland University of Technology / AIS Electronic Library (AISeL), Australia, pp. 1-16.

This file was downloaded from: <https://eprints.qut.edu.au/43815/>

© Copyright 2011 please consult authors

This work is covered by copyright. Unless the document is being made available under a Creative Commons Licence, you must assume that re-use is limited to personal use and that permission from the copyright owner must be obtained for all other uses. If the document is available under a Creative Commons License (or other specified license) then refer to the Licence for details of permitted re-use. It is a condition of access that users recognise and abide by the legal requirements associated with these rights. If you believe that this work infringes copyright please provide details by email to qut.copyright@qut.edu.au

Notice: *Please note that this document may not be the Version of Record (i.e. published version) of the work. Author manuscript versions (as Submitted for peer review or as Accepted for publication after peer review) can be identified by an absence of publisher branding and/or typeset appearance. If there is any doubt, please refer to the published source.*

<http://aisel.aisnet.org/pacis2011/184/>

9 July 2011

The Impact Of Software-As-A-Service On Business Models Of Leading Software Vendors: Experiences From Three Exploratory Case Studies

Sebastian Stuckenberg

University of Mannheim, stuckenberg@uni-mannheim.de

Erwin Fielt

Queensland University of Technology, e.fielt@qut.edu.au

Timm Loser

University of Mannheim, loser@wi.uni-mannheim.de

ISBN: [978-1-86435-644-1]; Full paper

Recommended Citation

Stuckenberg, Sebastian; Fielt, Erwin; and Loser, Timm, "The Impact Of Software-As-A-Service On Business Models Of Leading Software Vendors: Experiences From Three Exploratory Case Studies" (2011). *PACIS 2011 Proceedings*. Paper 184.
<http://aisel.aisnet.org/pacis2011/184>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2011 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

THE IMPACT OF SOFTWARE-AS-A-SERVICE ON BUSINESS MODELS OF LEADING SOFTWARE VENDORS: EXPERIENCES FROM THREE EXPLORATORY CASE STUDIES

Sebastian Stuckenberg, Department of Information Systems, University of Mannheim, Mannheim, Germany, stuckenberg@uni-mannheim.de

Erwin Fielt, Information Systems Discipline, Queensland University of Technology, Brisbane, Australia, e.fielt@qut.edu.au

Timm Loser, Department of Information Systems, University of Mannheim, Mannheim, Germany, loser@wi.uni-mannheim.de

Abstract

The number of software vendors offering ‘Software-as-a-Service’ has been increasing in recent years. In the Software-as-a-Service model software is operated by the software vendor and delivered to the customer as a service. Existing business models and industry structures are challenged by the changes to the deployment and pricing model compared to traditional software. However, the full implications on the way companies create, deliver and capture value are not yet sufficiently analyzed. Current research is scattered on specific aspects, only a few studies provide a more holistic view of the impact from a business model perspective. For vendors it is, however, crucial to be aware of the potentially far reaching consequences of Software-as-a-Service. Therefore, a literature review and three exploratory case studies of leading software vendors are used to evaluate possible implications of Software-as-a-Service on business models. The results show an impact on all business model building blocks and highlight in particular the often less articulated impact on key activities, customer relationship and key partnerships for leading software vendors and show related challenges, for example, with regard to the integration of development and operations processes. The observed implications demonstrate the disruptive character of the concept and identify future research requirements.

Keywords: Software-as-a-Service, business models, impact

1 INTRODUCTION

Software-as-a-Service is currently enjoying great popularity in research as well as in practice. An increasing number of software vendors opt to implement the concept and offer their solutions in a Software-as-a-Service mode. The solutions are extended and complemented by various partner companies, offering a complete ecosystem of software and secondary services to potential customers. The new concept, however, differs fundamentally from existing approaches of software delivery and therefore questions existing business models and industry structures. Delivering services directly to the customers raises questions, like whether the software vendor's relation with the customer and the role of the customer in the development and delivery processes are affected. Or how partner companies, like integrators or implementation consultancies react on the direct competition and on the possibility of their current business models to become obsolete? These and similar questions arise and motivate further research, as the full scope of implications of Software-as-a-Service on the rational of how an organization creates, delivers and captures value is not analyzed yet. Current research is scattered on focused aspects, like adoption factors and risks. Only a few studies with a more holistic view of the impact from a business model perspective exist. As normally implications are not focused on single aspects of a business model and interdependencies between multiple domains may exist, a more general observation of implications of the concept, however, seems promising.

One can expect that the Software-as-a-Service business model will be dependent upon a number of factors, like the kind of software (e.g. software for individuals or software for organizations) and the type of company (e.g. leading software vendors or new start-ups). To scope our research we will initially focus on the business models of leading software vendors, that are providing complex software for organizations (e.g. ERP, CRM, etc.) and are active on the Software-as-a-Service market, either as pure play (only Software-as-a-Service) or as hybrid (Software-as-a-Service next to a traditional software product) providers.

Based upon the discussion above, the following research questions have been identified that will be addressed in this study: *How does the Software-as-a-Service concept influence the business models of leading software vendors?* Or in other words, *what implications of the Software-as-a-Service concept for the business model can be identified?*

The business model ontology of Osterwalder and Pigneur (2010) provides the frame to structure this research. The analysis is initially focusing on distinct components of this structural aid (table 1). Subsequently the research questions will be discussed with an overall picture of business models in mind. For each component the current literature will be presented at first, before the discussion is extended with findings of explorative case studies of leading vendors with Software-as-a-Service offerings. The findings are taken as a preliminary step to gain first and still limited insights into the concept's implications on Software-as-a-Service vendors' business models.

The remaining sections of this paper are organized as follows: Section 2 presents foundations of Software-as-a-Service and business models, as well as a literature review of identified implications of the concept on business models. Section 3 introduces the conducted case studies and the research methodology, followed by key findings of the cases in section 4. The paper ends with a discussion in section 5 and the conclusion.

2 LITERATURE REVIEW

In order to evaluate the broad question of the Software-as-a-Service concept's implications on the way organizations, like software vendors, create value in their day-to-day operation, business models and the corresponding literature offer promising structural methods to reduce the complexity. At the same time it is important to process a shared understanding of the Software-as-a-Service concept and its characteristics. Therefore, Software-as-a-Service and business models will first be introduced and defined. Thereafter, previous research related to business models for Software-as-a-Service will be discussed.

2.1 Software-as-a-Service

The common understanding of the Software-as-a-Service concept has matured over the last years. There are still varying definitions used but the key criteria of the concept are shared. Mäkilä et al. (2010) empirically tested seven criteria and showed that although Software-as-a-Service vendors themselves apply different configurations, the core idea however is found in all companies.

In general Software-as-a-Service can be seen as a software distribution model of software vendors that differentiates itself from traditional approaches, like shrink-wrapped software in three main areas:

1. *Service property*: Services can be defined using three constitutive criteria: immateriality, the un-actu-principle and the existence of an external factor within the fulfilment phase. The Software-as-a-Service concept complies with these three criteria (Stuckenberg and Heinzl 2010), resulting – as common for services – in a continuous (service) relationship between the Software-as-a-Service vendor and the customer.
2. *Deployment model*: The altered deployment model is a direct consequence of the service property. Software-as-a-Service solutions are operated and maintained by the software vendor itself (Xin and Levina 2008). The vendor takes over activities that previously lived within the responsibility of the customer and extends its area of responsibility beyond development activities to operating and maintaining. Typical characteristics and technical implementations linked to this are the use of a multi-tenancy architecture (Aulbach et al. 2008), the access of the software using web-browsers (Sääksjärvi et al. 2005), and the adoption of service-oriented architectures (Turner et al. 2003).
3. *Pricing model*: The third differentiation point is the pricing model that is based on the continuous service relationship between the customer and the vendor. Deviating from the traditional model of licenses, the Software-as-a-Service pricing schemes are based on time or usage dependent metrics (Sääksjärvi et al. 2005). The previously charged continuous maintenance and support fees are in general embedded in the subscription fee (Cusumano 2008). The duty of payment persist as long as a company is subscribed to the service.

2.2 Business Models

Every company has a business model, whether that model is explicitly articulated or not (Chesbrough 2006; Teece 2010). Business models matter; the same idea or technology taken to market through two different business models will yield two different economic outcomes (Chesbrough 2010). Business models are required because of the features of market economies where there is consumer choice, transaction costs, heterogeneity amongst consumers and producers, and competition (Teece 2010). Ghaziani and Ventresca (2005) concluded that the business model discourse is mostly framed around value creation. Business models describe ‘the rationale of how an organization creates, delivers and captures value’ (Osterwalder and Pigneur 2010). The definition is well aligned with other definitions, such as from Chesbrough (2006), Johnson (2010) and Teece (2010).

A business model framework describes the compositional elements what a business model is made of. The elements are for example also referred to as components (e.g., Pateli and Giaglis 2004), (key) questions (e.g., Morris et al. 2005), or functions (e.g., Chesbrough and Rosenbloom 2002). Business model frameworks do not only define the elements, they also define the relationships between the elements (e.g., Gordijn et al. 2005). One of the more prominent and recent frameworks is the *Business Model Canvas* from Osterwalder and Pigneur (Fielt 2011). We focus on the framework of Osterwalder and Pigneur (2010) to structure our research, because it is widely applied and supported (for example, it was co-developed with 470 practitioners) and it is based on the already well-founded ontology of Osterwalder (2004). It is also generic enough to reflect the Software-as-a-Service as well as the traditional on-premises business models. The *Business Model Canvas* presents a shared language for describing, visualizing, assessing and changing business models. The nine building blocks of the model are summarized in table 1. The ontology offers a clear structure and allows to study business models at different levels of detail. It therefore promises to be a valid means for structuring the exploratory case study on the influence of Software-as-a-Service on business models,

reducing the complexity by discussing the findings first separately along the different building blocks and subsequently in an integrated, holistic perspective.

<i>Building Block</i>	<i>Description</i>
Customer Segments	The <i>Customer Segments</i> building block defines the different groups of people or organizations an enterprise aims to reach and serve. In order to better satisfy customers, a company may group them into distinct segments with common needs, common behaviours, or other attributes. An organization must make a conscious decision about which segments to serve and which segments to ignore.
Value Proposition	The <i>Value Proposition</i> building block describes the bundle of products and services that create value for a specific <i>Customer Segment</i> . It solves a customer problem or satisfies a customer need.
Channels	The <i>Channels</i> building block describes how a company communicates with and reaches its <i>Customer Segments</i> to deliver a <i>Value Proposition</i> . <i>Channels</i> comprise the customer interface with touch points that play an important role in the customer experience.
Customer Relationship	The <i>Customer Relationships</i> building block describes the types of relationships a company establishes with specific <i>Customer Segments</i> . A company should clarify the type of relationship it wants to establish with each <i>Customer Segment</i> .
Revenue Streams	The <i>Revenue Streams</i> building block represents the cash a company generates from each <i>Customer Segment</i> . A company must ask itself, for what value is each <i>Customer Segment</i> truly willing to pay?
Key Resources	The <i>Key Resources</i> building block describes the most important assets required to make a business model work. These assets allow an enterprise to create and offer a <i>Value Proposition</i> , reach markets, maintain relationships with <i>Customer Segments</i> , and earn revenues.
Key Activities	The <i>Key Activities</i> building block describes the most important things a company must do to make its business model work. Like <i>Key Resources</i> , they are required to create and offer a <i>Value Proposition</i> , reach markets, maintain <i>Customer Relationships</i> , and earn revenues.
Key Partnerships	The <i>Key Partnerships</i> building block describes the network of suppliers and partners that make the business model work. Companies create alliances to optimize their business models, reduce risk, or acquire resources.
Cost Structure	The <i>Cost Structure</i> describes all costs incurred to operate a business model. Creating and delivering value, maintaining <i>Customer Relationships</i> , and generating revenue all incur costs. Such costs can be calculated relatively easily after defining <i>Key Resources</i> , <i>Key Activities</i> , and <i>Key Partnerships</i> .

Table 1. The nine business model building blocks of Osterwalder and Pigneur (2010)

2.3 Previous Research

The evaluation of the Software-as-a-Service business model is not entirely new. Especially in recent years, there have been a number of publications analysing the impact and change evoked by the Software-as-a-Service concept at different levels of analysis, for example on the software industry (Cusumano 2010), business models (Ma 2007), or on focused aspects/components of the business model. In the following we will discuss some of the recent contributions in particular in relation to leading software providers with complex business applications, structured using the different building blocks of the introduced business model framework of Osterwalder and Pigneur (table 1) but starting with an additional general perspective.

Heart et al. (2010) study the vendors' readiness to deliver Software-as-a-Service. The questioned companies perceive Software-as-a-Service as an evolutionary rather than a revolutionary innovation and therefore do not employ radically new business processes. The analysed data however indicate a mismatch between perceived and indirectly measured capabilities and therefore question this classification. This is further supported by the SAP employees Bandulet et al. (2010), who regard Software-as-a-Service as a disruptive innovation in the enterprise application market because the

concept fulfils the characteristics of disruptive innovations defined by Christensen (1997). In the case Software-as-a-Service is regarded as a disruptive or revolutionary innovation, changes not only to limited aspects of a business model, like a new pricing model, but also to the business models as a whole can be expected. Disruptive innovations challenge existing models and processes and ask for existing knowledge to be re-evaluated and adapted to the new setting (Christensen 1997).

On an industry level, Hilker et al. (2010) study the implications of the “as-a-Service”-paradigm and compare two software ecosystems with CRM offerings. They assume that the paradigm shift will result in changes for all involved market players. They expect an intensified competition, with customers of Software-as-a-Service offerings being able to choose from a wider variety of extensions at lower prices. Due to the declining role of integrators, costs for integration and configuration will decrease.

In a broader context, Software-as-a-Service is positioned as part of Cloud Computing and represents the software layer in the architecture stack model. It is based on middleware and the underlying hardware. If offered to customers, the lower architectural levels are referred to as Platform-as-a-Service and Infrastructure-as-a-Service respectively. Weinhardt et al. (2009) offer a classification of different business models at the different architectural layers. They conclude that new business models and related research is required.

Customer Segments: The current literature does not reveal any clear indications regarding the targeted customers. Characteristics of Software-as-a-Service adopting organizations have been studied including companies of all sizes. There seems to be a trend that Software-as-a-Service is especially suitable for small and medium sized companies, however also small corporations are among the adopters (Benlian and Hess 2009; Benlian and Hess 2010). The market is further widened by the emergence of Software-as-a-Service platform accompanying extension marketplaces that allow vendors an easier access to a global market (Hilker et al. 2010). Additional revenues may also be generated from customer segments, previously not able to afford complex solutions (Anding 2010).

Value Proposition: Custumano (2008) takes a general perspective and transfers the trend to move towards service revenues in other industries to the software industry. Software-as-a-Service is not the focus of his considerations nor accurately matched in his collected data, but seen as a part of the service trend. A value-adding ‘servitization’ of products, and Software-as-a-Service may be seen as a service version of a software product in this context, is a major challenge for future business models of software vendors. Aspects that are especially relevant in the context of the *Value Proposition*. With bundles of software and services, Software-as-a-Service vendors can differentiate their offerings from traditional software vendors and offer the advantage of lower implementation costs (Fan et al. 2009). In addition to the core service, supporting services gain importance to accelerate the transition towards the service model (Mohammed et al. 2010). Part of the value proposition of a vendor is furthermore, that it offers to automatically keep the software updated and therefore provides the customer with the latest state of development (Saeed and Jaffar-Ur-Rehmann 2005). This implies fundamental changes to release-based development and versioning practises (Olsen 2006).

Channels: The prevalent consequence with respect to *Channels* is the use of the internet and related technologies as the primary delivery channel (Sääksjärvi et al. 2005). Apart from the core service delivery the internet is also used as major communication channel regarding information provisioning, training or support activities, facilitating web collaboration platforms like webcast or forums (Stuckenberg and Heinzl 2010).

Customer Relationship: The accumulated responsibility of the complete architecture stack in the role of the Software-as-a-Service vendors strengthens its position in the industry. The customer channel is owned by the vendor and providers of lower architectural layers are not required to maintain a communication channel with the customer anymore (Anding 2010).

Lower switching costs will lead to increased investments in customer loyalty (Hilker et al. 2010). An example are higher investments into software development with the result of an increased software quality that continuously convinces the customer to remain with the vendor (Choudhary 2007). Furthermore, a closer customer relationship in phases like requirements engineering is expected, as a

consequence of the hybrid software and service character (Berkovich et al. 2010). Analogously does the direct customer relationship make additional user and interaction related information available, that can be facilitated to increase the customers satisfaction or further intensify the communication (Saeed and Jaffar-Ur-Rehmann 2005).

Revenue Streams: Cusumano (2008) expects service based revenues to substitute product revenues and the price for standardized products to drop dramatically. His data reveal product revenues to account for only 50% of software companies' revenues in 2003. The profitability of software vendors seems to be positively influenced if service revenue takes up below 20% or more than 60% of the total revenues. The potential new customer segment, that is opened up by the reduced complexity and smaller up-front investment requirements, may generate further revenues, however, is not expected to offset missing license revenues (Anding 2010). Apart from the pure subscription fees, costs or charges for existing contracts may play an important role, as they create a lock-in effect but at the same time adoption barriers (Ma 2007). In addition, supporting services are supposed to account for a major share of profits and costs (Mohammed et al. 2010).

With a mature pricing model that evaluates fine grained software usage, the transaction volume and volatility of customers is influencing the competitiveness of Software-as-a-Service to traditional solutions. In such a context, decreasing software quality is asked to be counter measured with increased prices (Ma and Seidmann 2008). However, a recent study has revealed that currently vendors do not apply actual utility pricing schemes and mainly use user-based, per-month subscription fees instead. Pricing therefore is still very similar to traditional license models (Gartner 2009).

Key Resources: The control of the complete architecture stack requires building up resources and capabilities on all layers. For current application software companies for example, especially infrastructure or ecosystem management related capabilities may not have been in the focus before (Anding 2010). The service availability and elasticity aspects, drive complexity of service provisioning and call for altered mindsets in development of the software and operation of the service (Benefield 2009). The capabilities to deliver Software-as-a-Service and especially these that are not directly adoption related are not addressed appropriately. Vendors currently seem to overestimate their capabilities and do not expect Software-as-a-Service to challenge existing practises (Heart et al. 2010).

Delivering Software-as-a-Service helps to protect the intellectual property of vendors and reduces the chance of competitors to reproduce resources. For example, software piracy is avoided as source code is not leaving the control of the vendor anymore (Saeed and Jaffar-Ur-Rehmann 2005).

Key Activities: *Key Activities* of Software-as-a-Service vendors are the software development, deployment and operations processes, since these are essential for the value creation of the company. In the service context the value creation is archived to a considerable degree in the later of these activities (Ramaswamy 1994). Significant changes to the processes are expected (Espadas et al. 2008; Saeed and Jaffar-Ur-Rehmann 2005; Stuckenberg and Heinzl 2010). Among the envisioned implications are continuous, integrated processes of development and operations, a changed quality understanding, or information quality improvements, gained from the direct end-user relationship (Stuckenberg and Heinzl 2010). The emergence of Software-as-a-Service accompanying platform offerings may for example predefine architectural or technological aspects and affect related evaluation and design activities (Espadas et al. 2008). The Software-as-a-Service concept at the same time moves the development focus towards criteria like reusability, scalability, and availability (La and Kim 2009). Commonly used technologies, like multi-tenancy, further support this change. The prioritization switches from feature enhancement to operational cost decreasing functionality (Aulbach et al. 2008). Berkovich et al. (2010) take a focused approach on the requirements engineering phase of the software development process and argue similarly with the emergence of a second stakeholder, the service provider, that requests operations related requirements. The requirements engineering methods therefore need to incorporate software and service aspects and domain knowledge, continuous processes with tight customer integration and be capable of multiple requirement sources and stakeholders.

Key Partnership: Due to the increasing number of partners that are not necessarily bound by contracts to the provider and instead more market-organized, Hilkert et al. (2010) expect a growing requirement of developing new skills in the management and orchestration of partner relationships. They argue with a higher degree of market coordination in “as-a-Service” ecosystems, since a coordination implementing a market would result in comparably lower absolute transaction costs in the relationship between provider and partners. Cusumano (2008) predicts an increased direct competition of partners and vendors regarding activities that previously were not addressed by vendors.

The business model of partners may differ from models of traditional software vendor partners. The role of the conventional integrator, responsible for composing and configuring software solutions or providing the market with information will decrease because major parts of these activities are taken over by the Software-as-a-Service vendors themselves. On the other side, trust building activities, legal counselling services or risk consulting may create new business opportunities (Hilkert et al. 2010). Cusumano (2008) calls these revenues a result of a platform switch.

Cost Structure: Software-as-a-Service vendors experience an increased cost pressure, as they have to compete with traditional vendors and invest in software quality to be competitive but at the same time handle operational costs. To achieve a sufficient perceived quality by the customer, high quality features and functions as well as efficient service operations, and therefore a competition on two fronts, is required. Both are significant cost drivers (Fan et al. 2009). Operational expenses replace capital expenses in the improvement priority list (Aulbach et al. 2008). The ‘productization’ of services, in the sense of standardising services for a more efficient delivery, is of equal importance as the ‘servitization’ of products (Cusumano 2008).

3 METHODOLOGY

This research takes an explorative multi-case study approach. It is particularly suitable for answering the research question of *how* the Software-as-a-Service concept influences current business models of software vendors opting for the approach. Since the objective is to identify general changes of business models and implications evoked by the implementation of the Software-as-a-Service concept, the unit of analysis is the company’s business model. The focus is initially set on leading vendors. To be able to allow more detailed observations regarding the *Channels* and *Key Partnership* components, not only employees of the Software-as-a-Service provider itself but also a selection of their partners was interviewed. The interviews had an average duration of around one and a half hours and were recorded and afterwards transcribed to allow a better analysis. The analysis was done using descriptive codes, structured according to the nine elements of the business model ontology proposed by Osterwalder and Pigneur (2010). New nodes were added to the coding scheme as new implications emerged during the analysis (Miles and Huberman 1994). For triangulation purposes, the expert interviews were complemented by publicly available documents, especially the companies’ internet pages (Yin 2009).

Case	Leading vendor in	Pure Software-as-a-Service vendor	Hybrid Software-as-a-Service and on-premises vendor	Architectural layer origin of interviewee	Number of Interviews and affiliation
Case A	Software-as-a-Service	X		Software	Employees: 2 Partners: 1
Case B	On-premises software		X	Software	Employees: 3 Partners: 0
Case C	On-premises software		X	Infrastructure	Employees: 2 Partners: 1

Table 2. Case characteristics

All of the selected organizations offer complex business applications as a Software-as-a-Service solution and can be considered as market leaders, either with their Software-as-a-Service or product offering which they now complement by a Software-as-a-Service variation. Complex business

applications in this context refer to software like customer relationship management systems or enterprise resource planning systems. As illustrated in table 2, two of the companies (B and C) also sell software in the traditional approach (on-premises) and have its origin in this area. Company A already extended its Software-as-a-Service solution with a platform offering, allowing third-parties to enhance the solutions by complementary applications. This can be regarded as an indication for a matured state of Software-as-a-Service concept implementation. In case A and B the interviewed experts' units and work profiles were strongly affiliated to the company's Software-as-a-Service offering. The job positions varied from senior sales consultants to technology experts. In case C the interview partners belonged to departments dealing with lower layers of the architectural stack model. This was expected to allow further insights into operations and infrastructure related implications and additionally to contrast Software-as-a-Service and Infrastructure-as-a-Service business models.

4 RESULTS

In the following section some of the implications of the Software-as-Service concept, that were mentioned during the interviews are highlighted.

Customer Segments: With regard to *Customer Segments*, no clear implications emerged. The reduced required up-front investments allow to target new customer groups that previously could not afford the offered solutions because they were too expensive or too complex to deploy. Target customers are the functional departments but with involvement of the IT departments. Times, when the IT department was bypassed are said to be over, as defining and implementing an aligned strategy concerning Cloud Computing related concepts and technologies is promoted by companies' IT divisions. Apart from the mentioned extension of possibly targeted customers, companies of all sizes and domains are addressed by Software-as-a-Service vendors. There is no clear focus on a specific customer segment. Quite the opposite can be observed, the market is increasingly global. The global reach is enabled for instance by platforms and integrated marketplaces.

“Due to the underlying technology, we do not have size limitations to specific customer segments. We also don't have a certain industry focus.” (Case A, employee 2)

Observed limitations are based on the objectives to protect existing product and product accompanying service revenues from being cannibalized by the new offering.

Value Proposition: Regarding the *Value Proposition*, it can be observed that the Software-as-a-Service concept markets a transparent price model that is geared to customers. Elastic capacity is one of the selling points. It allows the dynamic assignment of resources like computing power or data storage. This for example makes it possible to generate real-time BI reports or master seasonal demand peaks without the need to reserve dedicated resources or run batch processes in periods of low system usage e.g. during the night. These capabilities are provided in an “as-a-Service” model exclusively.

The Software-as-a-Service concept changes the structure of the software industry because customers receive the complete software stack from one vendor, who gets responsible for orchestrating the underlying architectural levels. Integration efforts between architectural layers are not the problem of the customer anymore. Related support requests significantly decrease. The value proposition is therefore more complete in satisfying the customer's wish for software to provide functionality to support his business, rather than undesirably increasing complexity by software handling related issues.

A major differentiator and change to traditional approaches is the possibility of a vendor to easily allow flexibility in the scope of the implementation of a customer.

“The whole implementation approach does not target on a Big Bang, instead [a customer] can start with small functionality and thank to the scalability of the software, regarding users, but also regarding the flexible unlocking of additional functions, [a vendor] can sell the software gradually.” (Case A, employee 1)

As a result implementation projects are getting smaller and especially initial project may be realized within shorter time frames. For the vendor this has the advantage of sales activities not binding as much long-term resources as in traditional approaches, which projects easily take several months. The general value proposition of software, that a customer can implement a business process with it, can be achieved more efficiently with a service model, as knowledge and experiences of required but no direct value creating activities of operating the software can be left to the vendor.

Channels: Software-as-a-Service vendors intensively use direct sales channels, like their websites, to attract new customers and initiate a sales process. Often additional documents and reports, that require the customer to provide his contact information prior to access or download, are offered and the collected data utilized to start further sales activities.

“A high percentage of our leads originate from our homepage.” (Case A, employee 2)

This is further supported by the popular possibility to offer trial periods to potential customers, that allow to extensively test the software with its full functionality without any further obligation or setup requirements. Initial projects are used to convince the customer and to generate further projects with additional revenues.

“Our aim is to get the customer productive as quick as possible and by proving our capabilities, be able to collect further requirements to be implemented subsequently.” (Case A, employee 2)

In development, product ideas are provided by the customer base or partners using idea platforms and are used to enhance the offering. Seamlessly integrated solutions are also used to provide help or to train the users. These functions go beyond static pages and include forums, videos and interactive tutorials.

“I don’t need to open a separate channel. In on-premises software, there is a media transfer. In on-demand it is much easier to offer help over the internet and to provide the customer with a much better overall experience.” (Case B, employee 1)

Customer Relationship: The partly new or intensified communication channels already indicate a changed customer relationship, which is also signalled by a few other interesting facets that emerged during the interviews. As already indicated by the direct sales channels, further aspect suggesting a closer and continuous relationship between the vendor and the customer can be observed.

“In fact, you are much closer to the customer and much closer to the customers’ problems, when you operate [the software] yourself.” (Case A, employee 2)

In the support departments, questions regarding installation specifications are obsolete. In fact the support staff can directly look into the meta information of an incident, is aware of the latest interactions with the system and can provide a quick solution. In the on-premises world this was limited by the high number of possible variations of configurations. Due to the multi-tenancy, the solution is immediately available to all customers. The usage/user related information can be further utilized to prioritize development activities, putting more resources on aspects that are used heavily. At the same time usage information may identify problems the users have when using the software.

“That means, I have the chance to develop the perfect application. This chance did not exist before, because I had to spend a lot of money to gather this data using traditional market research techniques.” (Case B, employee 2)

On a more general level the information gains can be facilitated to increase customer satisfaction.

“Because with the system operation, we have a good insight into how much a customer works with the system. We can’t see what he does, but we can see if he uses it. And an indicator for an unsatisfied customer is one, who doesn’t use the system that frequently.” (Case A, employee 1)

A closer customer relationship is further expressed by the requirement to have an increase awareness of customers’ processes and problems.

“It is called impact management. It already exists in the hosting area, you for example know when your customer runs critical processes, like the end-of-quarter closing, and take special care of the customer during this times.” (Case B, employee 1)

Revenue Streams: Related to the *Revenue Streams*, the interview partner mentioned the typical advantages for customers. The move towards variable instead of fix costs, as a result of the usage or time based subscription fees. In addition vendors can charge for the additional activities that were previously done by the customers themselves. However, the cases did not reveal further implications for Software-as-a-Service vendors.

Key Resources: Building up competence regarding infrastructure and operation activities is considered as an important implication of the Software-as-a-Service concept. These are required to drive innovation and assure an efficient service delivery at manageable costs. Being responsible for the operation of the service also affects the core competencies of the vendors, as security aspects are now perceived as key capabilities. But also competence regarding technologies and methods that allow to efficiently scale is highly demanded.

"We are a company that considers IT security as its core competence. We think of IT operations as our core competence." (Case B, employee 2)

Key Activities: The most obvious implication is of course the extension of the process of value creation with operation related activities. Parts of these latter steps, especially the process to get a customer using the system is highly standardized and tool-supported. Perceived complexity increases exist with regard to update cycles, as updates or enhancements affect the whole customer base at once, due to the central multi-tenancy architectures. A disruption of customers' processes, however, has to be avoided.

"I can't just add new functions and the user does not recognise his work environment on the next day anymore. Enhancements have to work in a complete different way." (Case B, employee 1)

Compatible continuous development is required and is challenging existing release-based development cycles. This is especially important, since the customer does not have the possibility to choose if he wants an update. The advantage for the vendor, however, is that it can focus all its resources on one release, without being required to reserve developers for the support of older versions. This is perceived to accelerate the development and innovation process. Innovation priorities move from functions to performance and cost efficiency.

Key Partnerships: The observations concerning partnerships can be distinguished between implications for the Software-as-a-Service vendor and those that affect the business model of the partners. The latter were not in the focus of this paper so far; however some findings are worth mentioning, since they have an indirect influence on the vendors' business model.

For the Software-as-a-Service vendor, the partner relationships are getting closer. Depending on the service of the partner, the two provided services are closer integrated and the requirement of both parties already during the conception phase is to be considered.

"Processes need to be closely aligned and integrated because you deliver a joined service. In the end the customer gets a solution that is as consistent as possible." (Case B, employee 1)

This implies that similar to the vendor, the customer relationships of partners are also aiming on long-term relations.

The Software-as-a-Service concept makes certain business models of partners redundant or let the vendor enter into a direct competition with value propositions of partners. One example is integration activities, which can be offered very easily by the vendor. Due to the central infrastructure approach, the vendor can integrate its solution with other applications once and that integration is afterwards available to all its customers. The previously local integration in every customer's system is not required anymore. Integrators specialized in these activities are left with small scale application integrations that are not accomplished by the vendors yet.

"The partners, whose business model target on making money from configuration or the infrastructure, won't exist anymore in the future." (Case A, employee 2)

On the other hand the concept also offers new revenue possibilities to partners. The changes require the partner to standardize and scale their services. The specific services need to be productized and offered to the global customer base of the Software-as-a-Service vendor.

“Partners need to use their specific added value knowhow to form new own products that they offer on market places and that are operated on platform solutions.” (Case B, employee 3)

One example in this context could be integration functions, which are offered to the customer as a service and link two Software-as-a-Service solutions together.

The implementation can, due to the standardization and accompanied acceleration of the process, be regarded as a productized, classic IT-service. It is charged as a fixed price and requires the partners to generate more volume by more implementation projects.

Cost Structure: Implications regarding the cost structure of Software-as-a-Service vendors are a consequence of the previously described changes. The focus on direct and shorter sales cycles for example lead to reduced costs for customers and vendors. However, the Software-as-a-Service concept and related technologies require a certain scale to be able to be operated at manageable costs and to take full advantage of its potentials. The investment risk for vendors is higher than before, because vendors have to make the specific investments and are exposed to a hold-up risk. The major implications of the Software-as-a-Service concept, however, are anticipated efficiency gains throughout the whole value chain.

“The bottom line is, that Software-as-a-Service as a deployment and operation mode is much more efficient than traditional on premise software, furthermore it allows to develop better software. For this reason, in the long run it has to be more profitable.” (Case A, employee 1)

5 DISCUSSION

The exploratory case studies as well as the literature review have revealed the impact of the Software-as-a-Service concept on the business models and practises of leading software vendors and raise new challenges in order to deliver the service successfully. Table 3 summarizes the findings and discussion on a business model element level. It shows that the Software-as-a-Service concept affects every building block and, thereby, confirms the starting-point of this paper. This underlines the need to assess the impact of Software-as-a-Service in an integrated, holistic manner by means of the business model and move beyond a focus on isolated aspects such as adoption factors or risks.

From a cross-element perspective the impact is driven by the customer-facing elements: it centres on the *Value Proposition* (the service concept) and it originates in particular from the *Channels* (delivery model) and *Revenue Streams* (pricing model) components. As the majority of the Software-as-a-Service concept definitions comprise the delivery and pricing model as key differentiating factors to the on-premises business, this is less surprising. More interesting though, are the implications of the residual, vendor-internal elements, as some of the identified consequences fundamentally challenge existing practises of leading software vendors. Based on the preliminary findings the *Key Activities*, *Customer Relationship* and *Key Partnership* aspects are particularly challenged. *Key activities*, as the value generation is not limited to software development anymore but also originated from an efficient service delivery involving all architectural layers. Existing development methodologies may not address the delivery aspects appropriately and cumber a tight integration of development and operations processes. The customer role is further strengthened, as the vendor establishes a long-term direct relationship when delivering Software-as-a-Service. It is required to response to customer wishes as it continuously needs to convince the customer to stay with the service. On the supply side, the Software-as-a-Service concept challenges parts of the existing structures of the partner ecosystem.

It can be observed that especially the customer/demand oriented elements (*Value Proposition*, *Channels*, *Revenue Streams* and *Customer Segments*) indicate similarities regarding the applied business model, e.g. with regard to the pricing scheme. The vendor/supply oriented elements vary more significantly. This is further supported by a focus of the Software-as-a-Service related discussion in the literature and in practise on how to foster customer adoption and solve related problems. Most attention is therefore attracted by the customer oriented components. It can however be speculated, that the main changes and challenges for leading software vendors concern supply oriented components like *Key Activities* and *Key Resources*. Supporting arguments for this hypothesis

can, for example, be found in current price models, as most of these are not yet applying real usage-based pricing schemes and are mainly based on fix time periods like monthly payments. A usage-based pricing model, that for example charges the number of times a certain function is used, may require changes to underlying internal processes or resources, like the software architecture, that are not yet addressed. A second innovation wave of customer/demand oriented components with new forms of *Value Propositions*, *Revenue Streams* or *Customer Segments* may be expected once the supply-related issues have been resolved, for example, introducing innovative functionality, extending customization options and new, needs-based customer segmentation.

<i>Building Block</i>	<i>Impact</i>
Customer Segments	Customer segments are manifold and an increasingly global market is addressed by vendors. Various adoption barriers and risks have been identified in the literature and it seems as if vendors themselves further support and propagate limitations because of their desire to protect existing product revenues.
Value Proposition	The value proposition promises flexibility and carefree service delivery with regard to resources, updates and enhancements, and even implementation scope. The literature and the cases, however, indicate that this requires fundamental changes to processes and mindsets, e.g. version-free software, to be realized.
Channels	Communication, distribution and sales channels are making extensive use of the Internet and vendors are utilizing subsequent opportunities, e.g. forums, idea platforms, or video streams to automate processes and to increase efficiency.
Customer Relationship	The customer relationship is expected to be closer, more direct and with a more continuous orientation. As a consequence of the direct feedback channel the information base regarding customer requirements is strengthened, however, the full potential is not yet utilized by the analysed vendors.
Revenue Streams	Revenue streams of Software-as-a-Service vendors are dependent on the variable pricing model of subscription fees and challenge existing practises of consulting or implementation services. The long-term consequences for and treatment of additional revenue streams are not yet clarified.
Key Resources	Infrastructure and operations related capabilities gain importance and extend the key resources of a software vendor.
Key Activities	The processes to develop, deploy and operate the service, as <i>Key Activities</i> , are highly affected and require to integrate development and operations activities and set a focus on efficiency in the whole end-to-end process. Existing release-based development cycles are challenged. The cases, however, left open whether vendors have addressed the resulting problems accordingly.
Key Partnerships	With regard to the key partnerships of a Software-as-a-Service vendor the focus of the discussion is on the competition between the vendors' and partners' business models, as part of the existing value propositions of partners are absorbed by the vendors. Whereas the literature takes up a rather high level perspective, the cases reveal concrete examples of conflicts but also point at opportunities for partners to engage in new fields. The competition is however only one aspect, since the service delivery increasingly requires a joint and consistent integration of vendor and partner processes. These somehow opposed objectives are yet rarely addressed in the discussion.
Cost Structure	As the last business model element, the resulting cost structure is pressurized by the additional operations related activities and obligation to investment in software as well as service quality. The regarded organizations, however, are convinced that the possible – maybe yet to address – efficiency gains will result in a profitable business model in the long-term.

Table 3. *Summary and discussion of implications on different business model components*

Concerning the research question, the previous discussion makes clear that the Software-as-a-Service concept strongly influences business models of Software-as-a-Service vendors but that the impact can differ, in particular for the vendor/supply oriented business model elements. The evaluation of leading vendors revealed a preliminary list of implications. Some of these have already been addressed by the vendors, others are yet to resolve.

To a certain extent the co-existence of both delivery models or the focus on a pure Software-as-a-Service model seem to have an impact, in particular on the *Customer Segments* and the *Key Partnerships*. Therefore, noteworthy differences between case A and B are triggered by the co-existence of both delivery models, Software-as-a-Service and on-premises, at organization B. The company implements protective measures and for instance artificially limits the customer segment according to the blank spots of the product business. Vendors with two business models may create sufficient differentiation factors to avoid conflicts with existing product revenues. Similarly, the partner ecosystem inherits possible disputes, as partners of the on-premises business may be in a situation of direct completion in the service model. However, offering software in a product as well as service mode may also create synergies with regard to resources or the cost structure, as the increased Software-as-a-Service related investment risks may be balanced with established revenue streams of the product business.

6 CONCLUSION

The Software-as-a-Service concept is expected to have major implications on the business logic of a software vendor in terms of how it creates, delivers and captures value when implementing the concept. Both literature and cases show that an integrated, holistic perspective by means of a business model is required to understand the full implications.

While most attention is focused on the impact on the value proposition, channels and revenue model, as they form the more visible part of the Software-as-a-Service concept, the impact on key activities, customer relationship and key partnerships of leading software vendors should not be underestimated. Some of the specific issues for leading software vendors related to these building blocks are the integration of development and operations activities, a focus on efficiency in the whole end-to-end process, closer, more direct relations with customers with a more continuous orientation, and a more balanced perspective of partners, as not only the possible threat of redundancies but also the need for closer collaboration and new revenue opportunities have to be considered.

The presented research results contribute to the existing body of knowledge, as they highlight implications and challenges of the Software-as-a-Service concept that are not yet or only limited addressed in current research. Potential research endeavours can be deducted. Among these, the internal perspective concerning the vendors' processes of value creation and delivery seems especially promising. For practice, the results point at potential problems vendors planning to enter the market with a Software-as-a-Service solution may encounter. They therefore create an awareness of potential conflicts and allow vendors to prepare accordingly.

The analysis was focused on leading software vendors with complex business applications. Therefore pricing models like free or freemium (Anderson 2009) and corresponding implications on revenue streams and other components were out of the scope of this first study. Leading vendors in addition may tend to regard the internal infrastructure, for example with respect to data centres as their core competency and key partnerships are therefore focusing on customer oriented activities, e.g. consultancies or integrators. The inclusion of smaller companies or start-ups with, for example, less financial power may yield different results as they most likely maintain close relations with partners providing resources, like processing power. *Key Partnerships* may thus have a different form.

Further research, including a broader set of organizations and Software-as-a-Service offerings in conjunction with a rigor and exhaustive literature review is required to address the mentioned limitations. In addition it appears promising to analyse possible patterns in the applied business models that result in Software-as-a-Service business model archetypes or business model classifications. These could help to evaluate general problems and implications of the concept.

Moreover, the current case study is limited in terms of number of cases and number of interviews. While this data showed the full implications in terms of the different business model elements, further interviews and cases will add to a more in-depth understanding of the implications and possible differences for different types and approaches of leading software vendors.

References

- Anderson, C. (2009). *Free: the future of a radical price*, Random House Business Books, London, UK.
- Anding, M. (2010). SaaS: A Love-Hate Relationship for Enterprise Software Vendors. In *Software-as-a-Service* (Benlian, A., Hess, T. and Buxmann, P. Ed.) Gabler, Wiesbaden, 43-56.
- Aulbach, S., Grust, T., Jacobs, D., Kemper, A. and Rittinger, J. (2008). Multi-tenant databases for software as a service: schema-mapping techniques. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1195-1206, ACM, Vancouver, Canada.
- Bandulet, F., Faisst, W., Eggs, H., Otyepka, S. and Wenzel, S. (2010). Software-as-a-Service as Disruptive Innovation in the Enterprise Application Market. In *Software-as-a-Service* (Benlian, A., Hess, T. and Buxmann, P. Ed.) Gabler, Wiesbaden, 15-29.
- Benefield, R. (2009). Agile Deployment: Lean Service Management and Deployment Strategies for the SaaS Enterprise. In *42nd Hawaii International Conference on System Sciences* (Sprague, R. H. Ed.), IEEE, January 5-8, Waikoloa, Hawaii.
- Benlian, A. and Hess, T. (2009). Welche Treiber lassen SaaS auch in Grossunternehmen zum Erfolg werden? Eine Empirische Analyse der SaaS-Adoptopn auf Basis der Transaktionskostentheorie. In *Wirtschaftsinformatik* (Hansen, R. H., Karagiannis, D. and Fill, H. G. Ed.), 567-576, February 25-27, Vienna, Austria.
- Benlian, A. and Hess, T. (2010). The risks of sourcing software as a service - An empirical analysis of adoptors and non-adopters. In *18th European Conference on Information Systems* (Alexander, T., Turpin, M. and van Deventer, J. Ed.), June 7-9, Pretoria, South Africa.
- Berkovich, M., Esch, S., Leimeister, J. M. and Krcmar, H. (2010). Towards Requiriements Engineering for Software as a Service, *MKWIGöttingen*, Germany.
- Chesbrough, H. (2006). *Open business models: How to thrive in the new innovation landscape*, Harvard Business School Press, Boston, MA.
- Chesbrough, H. (2010). Business Model Innovation: Opportunities and Barriers, *Long Range Planning*, 43 (2-3), 354-363.
- Chesbrough, H. and Rosenbloom, R. S. (2002). The role of the business model in capturing value from innovation: Evidence from Xerox Corporation's technology spin-off companies, *Industrial and Corporate Change*, 11 (3), 529-555.
- Choudhary, V. (2007). Software as a Service: Implications for Investment in Software Development. In *40th Hawaii International Conference on System Sciences* (Sprague, R. H. Ed.), IEEE, January 3-6, Waikoloa, Hawaii.
- Christensen, C. M. (1997). *The Innovator's Dilemma*, Harper Paperbacks, Boston, USA.
- Cusumano, M. A. (2008). The Changing Software Business: Moving from Products to Services, *Computer*, 41 (1), 20-27.
- Cusumano, M. A. (2010). Will SaaS and Cloud Computing become a new Industry Platform? In *Software-as-a-Service* (Benlian, A., Hess, T. and Buxmann, P. Ed.) Gabler, Wiesbaden, 3-13.
- Espadas, J., Concha, D. and Arturo, M. (2008). Application Development over Software-as-a-Service Platforms. In *The Third International Conference on Software Engineering Advances* (Mannaert, H., Ohta, T., Dini, C. and Pellerin, R. Ed.), 97-104, October 26-31, Sliema, Malta.
- Fan, M., Kumar, S. and Whinston, 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.
- Fielt, E. (2011). Understanding business models, *Business Service Management Whitepaper*, Vol. 3 Smart Services CRC, Eveleigh, NSW, Australia.
- Gartner (2009). *Fact Checking: The Five Most-Common SaaS Assumptions* Gartner.
- Ghaziani, A. and Ventresca, M. (2005). Keywords and cultural change: Frame analysis of Business Model public talk, 1975–2000, *Sociological Forum*, 20 (4), 523-559.
- Gordijn, J., Osterwalder, A. and Pigneur, Y. (2005). Comparing two business model ontologies for designing e-business models and value constellations. In *Proceedings of the 18th Bled*

- Electronic Commerce Conference* (Vogel, D. R., Walden, P., Gricar, J. and Lenart, G. Ed.) University of Maribor, Bled, Slovenija.
- Heart, T., Tsur, N. S. and Pliskin, N. (2010). Software-as-a-Service Vendors: Are They Ready to Successfully Deliver? In *Global Sourcing of Information Technology and Business Processes*, Vol. 55 (Oshri, I. and Kotlarsky, J. Ed.) Springer Berlin, Heidelberg, 151-184.
- Hilkert, D., Wolf, C. M., Benlian, A. and Hess, T. (2010). The "as-a-Service"-Paradigm and its Implications for the Software Industry – Insights from a Comparative Case Study in CRM Software Ecosystems. In *First International Conference on Software Business (ICSOB 2010)*, June 21-23, Jyväskylä, Finland.
- Johnson, M. W. (2010). *Seizing the white space: Business model innovation for growth and renewal*, Harvard Business Press, Boston, MA.
- La, H. and Kim, S. (2009). A Systematic Process for Developing High Quality SaaS Cloud Services. In *Cloud Computing* (Jaatun, M., Zhao, G. and Rong, C. Ed.) Springer Berlin, Heidelberg, 278-289.
- Ma, D. (2007). The Business Model of "Software-As-A-Service", *IEEE International Conference on Services Computing (SCC 2007)* Salt Lake City, USA.
- Ma, D. and Seidmann, A. (2008). The Pricing Strategy Analysis for the "Software-as-a-Service" Business Model. In *Grid Economics and Business Models* (Altmann, J., Neumann, D. and Fahringer, T. Ed.) Springer Berlin, Heidelberg, 103-112.
- Mäkilä, T., Järvi, A., Rönkkö, M. and Nissilä, J. (2010). How to Define Software-as-a-Service – An Empirical Study of Finnish SaaS Providers. In *Software Business* (Tyrväinen, P., Jansen, S. and Cusumano, M. A. Ed.) Springer, Berlin, Heidelberg, 115-124.
- Miles, M. B. and Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook*, Sage Publications, Thousand Oaks, CA.
- Mohammed, A. B., Altmann, J. and Hwang, J. (2010). Cloud Computing Value Chains: Understanding Businesses and Value Creation in the Cloud. In *Economic Models and Algorithms for Distributed Systems* (Neumann, D., Baker, M., Altmann, J. and Rana, O. Ed.) Birkhäuser, Basel, 187-208.
- Morris, M., Schindehutte, M. and Allen, J. (2005). The entrepreneur's business model: Toward a unified perspective, *Journal of Business Research*, 58 (6), 726-735.
- Olsen, R. (2006). Transitioning to Software as a Service: Realigning software engineering practices with the new business model. In *IEEE International Conference on Service Operations and Logistics, and Informatics*, 266-271, IEEE, June 21 - 23, Shanghai, China.
- Osterwalder, A. (2004). *The Business Model Ontology: A proposition in a design science approach*, University of Lausanne, Lausanne, Switzerland.
- Osterwalder, A. and Pigneur, Y. (2010). *Business model generation: A handbook for visionaries, game changers, and challengers*, (self-published).
- Pateli, A. G. and Giaglis, G. M. (2004). A research framework for analysing eBusiness models, *European Journal of Information Systems*, 13 (4), 302-314.
- Ramaswamy, R. (1994). *Design and management of service processes: Leaping customers for Life*, AddisonWesley, Reading, MA, USA.
- Sääksjärvi, M., Lassila, A. and Nordström, H. (2005). Evaluating the Software as a Service Business Model: From CPU Time-Sharing to Online Innovation Sharing. In *IADIS International Conference e-Society* (Isaisas, P., Kommers, P. and Mc-Pherson, M. Ed.), 177-186, June 27 - 30, Qawra, Malta.
- Saeed, M. and Jaffar-Ur-Rehmann, M. (2005). Enhancement of software engineering by shifting from software product to software service. In *First International Conference on Information and Communication Technologies (ICICT)* (Khan, W. and Ahmed, F. Ed.), 302-308, August 27-28, Karachi, Pakistan.
- Stuckenberg, S. and Heinzl, A. (2010). The Impact of the Software-as-a-Service Concept on the Underlying Software and Service Development Processes. In *14th Pacific Asia Conference on Information Systems*, 1297-1308, Taipei, Taiwan.
- Teece, D. J. (2010). Business Models, Business Strategy and Innovation, *Long Range Planning*, 43 (2-3), 172-194.

- Turner, M., Budgen, D. and Brereton, P. (2003). Turning software into a service, *Computer*, 36 (10), 38-44.
- Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinl, T., Michalk, W. and Stöber, J. (2009). Cloud Computing – A Classification, Business Models, and Research Directions, *Business & Information Systems Engineering*, 1 (5), 391-399.
- Xin, M. and Levina, N. (2008). Software-as-a Service Model: Elaborating Client-Side Adoption Factors. In *29th International Conference on Information Systems* (Boland, R., Limayem, M. and Pentland, B. Ed.), December 14 - 17, Paris, France.
- Yin, R. K. (2009). *Case Study Research - Design and Methods*, Sage Publications, Thousand Oaks.