7-15-2012

Software-As-A-Service: Implications For Business And Technology In Product Software Companies

Austin D'souza
Innovation Management & Strategy, University of Groningen, The Netherlands, a.dsouza@rug.nl

Jaap Kabbedijk
Information and Computing Science, Utrecht University, The Netherlands, j.kabbedijk@uu.nl

DongBack Seo
Innovation Management & Strategy, University of Groningen, The Netherlands, d.seo@rug.nl

Slinger Jansen
Information and Computing Science, Utrecht University, The Netherlands, s.jansen@uu.nl

Sjaak Brinkkemper
Information and Computing Science, Utrecht University, The Netherlands, s.brinkkemper@uu.nl

SOFTWARE-AS-A-SERVICE: IMPLICATIONS FOR BUSINESS AND TECHNOLOGY IN PRODUCT SOFTWARE COMPANIES

Austin D'souza, Innovation Management & Strategy, University of Groningen, The Netherlands, a.dsouza@rug.nl

Jaap Kabbedijk, Information and Computing Science, Utrecht University, The Netherlands, j.kabbedijk@uu.nl

DongBack Seo, Innovation Management & Strategy, University of Groningen, The Netherlands, d.seo@rug.nl

Slinger Jansen, Information and Computing Science, Utrecht University, The Netherlands, s.jansen@uu.nl

Sjaak Brinkkemper, Information and Computing Science, Utrecht University, The Netherlands, s.brinkkemper@uu.nl

Abstract

Many software organizations are currently transitioning from an on-premises deployment model to the Software as a Service (SaaS) model. If a company restricts changes only in the business or technical perspective, the transition leads to higher costs, poor adoption of the SaaS model, and in the worst case, the company can lose its business. Much literature focuses on changes within one domain and is generally also limited to one perspective. This paper provides stakeholders (i.e. product managers, and business managers) an integrated perspective (business and technological) with a comprehensive framework that covers changes in four domains: business/product structure, revenue logic, customer relationships, and partnerships. The applicability of the proposed framework is assessed with a case study of a large software product vendor.

The paper also contributes by providing a new avenue to study SaaS, with an integrated perspective for the organizational transition period. For the industry, this paper suggests a way to assess the impacts of organizational transition towards the SaaS model. With this overview in hand, software-producing organizations can use the comprehensive framework to successfully transition to become SaaS vendors.

Keywords: Software-as-a-Service, On-premises Software, Software Vendors, Business Models Transition, Deployment Models Transition

1 INTRODUCTION

There is a growing trend of software companies adopting the Software as a Service (SaaS) business model (Kaplan, 2005), where software is no longer deployed on-premises, but remotely available through the Internet instead. Gartner predicts that the SaaS related industry would reach about $21.3 billion by 2015 (Stamford, 2011). All kinds of software producing organizations are scrambling for a piece of the action, such as Microsoft's Office 365 and Google's Apps for Business.

Traditionally, a software solution is produced by a software vendor and shipped to a customer to be deployed on premises. This way of deploying software is referred to as *on-premises deployment*. The business model related to this approach is referred to as the *on-premises business model*. The costs and management overheads related to the acquisition and maintenance of IT infrastructure, including hardware, software and expertise, can be substantial for customers. Because of these high costs associated with on-premises deployment, the SaaS business model is increasingly being adopted. Other reasons for SaaS being so popular among software organizations and their customers are, amongst others, the increased time to market and a better customer relationship (Albach, Grust, Jacobs, Kemper & Rittinger, 2008). In the *SaaS business model*, the ownership and management of a software product is outsourced to a software provider through the Internet. The deployment method related to the SaaS business model is referred to as *SaaS deployment*. This shift from the on-premises model to the SaaS model has consequences from both the business and technological perspectives.

Making the transition to SaaS model is not easy, especially for companies who have a long history and a large amount of infrastructure built to develop and deliver software under the on-premises business model. These companies are used to a steady influx of revenue due to new sales, which will be much lower in the case of SaaS, where revenue is spread over time and contract value in the first year tends to be significantly lower, even when total revenue per customer will be higher in the end. The frustration and difficulty felt by managers is best summarized by the quote below.

"[The transition to SaaS] is a hard transition. It was a hard transition for us to accept a couple of years ago, and it's one that's full steam ahead at Microsoft. [SaaS] is a different business model" - Microsoft COO Kevin Turner (Cowley & McLaughlin, 2007).

The contribution of this paper to the body of the software as a service literature is twofold; first, it provides an integrated perspective and focus domains for researchers and software companies to consider in studying and moving their businesses toward Software as a Service (SaaS). Second, based on the analysis of these domains, recommendations are made for researchers and practitioners who are involved in transitioning a business from on-premises to SaaS.

2 RESEARCH METHOD

The goal of this paper is to study the changes taking place in software producing organizations' business models and the technical consequences of this change during the transition from an on-premises model to a SaaS model. Hence qualitative research method is adopted as it is best suited to comprehensively describe, understand, and explain the complexity and dynamics of management phenomenon at organizational level (Delattre, Ocler, Moulette & Rymeyko, 2009). Four process change domains are identified, based on a literature review and pilot interviews with two academic researchers from universities and three practitioners from software companies. Especially, according to Rajala, Rossi and Tuunainen (2003), the four core elements of a business models (Valtakoski and Rönkkö, 2010) are product strategy, revenue logic, distribution model, and service and implementation model. Considering the goal of this paper, the term product strategy has been renamed to Product/Business structure to encompass both the technical and business perspectives and it also includes the concept of cost structures. Since the challenge within the SaaS model is not so much about how to distribute or implement the product, but it is to maintain customer relationships, we have focused on changes in customer relationships instead of distribution model. The service and implementation model is renamed to changes in partnerships, because a company establishes and maintains partnerships to provide valuable services to customers without implementing products on customers' sites in the SaaS environment. Therefore, the four domains are identified and named as: i) changes in business/product structure, ii) changes in revenue logic, iii) changes in customer relationships, and iv) changes in partnerships. Then, these four domains are applied to a case study at a software company, which is changing its business model towards SaaS to confirm the applicability of the process changes domain model.

The case study method follows the guidelines of Yin (2009). The researchers have kept a case study database and have discussed key findings with case study participants. A total of 12 semi-structured interviews of about 2 hours each were conducted with both technical and business experts at the company, and they have studied internal documents (i.e. product brochures, technical documentation, customer contracts) and publically available web sites. After the data was transcribed and categorised based on the four process domains it was cross verified with the respondents for accuracy. Based on this data source the conclusions for this research were formed. The case was selected because the company is a frontrunner in the Netherlands in regards to software development and because the company has invested major resources in the transition project.

3 REVIEW OF ON-PREMISES BUSINESS AND DEPLOYMENT MODELS

3.1 On-premises business model

On-premises software business models can be broadly categorized as product tailor and product licensor (Kontio, Jokinen, Mäkelä and Leino, 2005).

Product tailors: Product tailors are companies who generate revenue based on product licenses.

However they customize software based on customer needs. They are also very closely involved with customers on an individual level (Kontio et al., 2005).

Product licensors: The most successful and dominant business model in the software industry is the product licensor (Popp, 2011). Hence the on-premises business model will be further explored in context of product licensor business model. The business model in its simplest form involves developing, selling, implementing, and managing the software. This can become complex depending on the type, scope, and complexity of the software. Companies usually develop and manage software as a product, regularly updating and releasing newer versions of the software, for example windows operating systems and ERP software. This type of software is highly productized and the product licensors involvement with the customer is low (Kontio et al., 2005). This type of software is generally licensed based on one or a combination of the following factors: number of users, modules, and number of installations (Mousavidin & Silva, 2009).

Figure 1 depicts a high level representation of the on-premises business model. Two of the most important resources used in software development are software development tools and knowledge. It is a common practise to acquire knowledge and capabilities through outsourcing partners (Rauscher and Smith, 1995). Software development tools include necessary development platforms, software, and hardware. Outsourcing partners provide necessary skills, capabilities, and knowledge that can aid in software development. Sales and marketing, software research and development, and customer support are important activities for software companies (Chou, 2005). Research and development mainly develops and manages on-premises software. Sales and marketing is responsible for selling software licenses (perpetual or for a limited period of time) and services, which is their main source of revenue. Customer services are responsible for providing services such as customers support and implementation services. Channel partners can sometimes participate in conducting these processes (Chou, 2005).

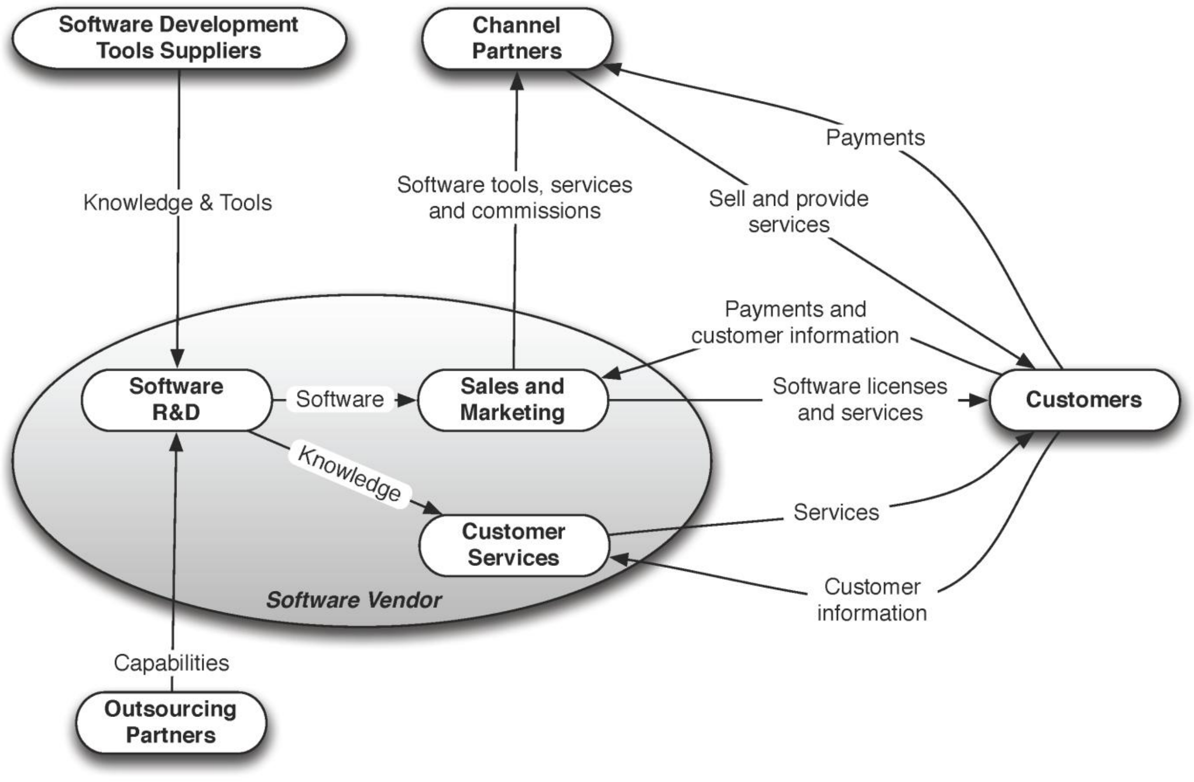

Figure 1. High level representation of on-premises software business model

In order to do so, they need to invest additional amounts of resources (e.g., money) on infrastructure (hardware, software & human resource) and managing the needed infrastructure (Jaekel & Luhn, 2010).

3.2 On-premises deployment method

In the on-premises software deployment method, the software is shipped to the customer after release, where it has to be installed (Carzaniga, Fuggetta, Hall, Heimbigner, van der Hoek & Wolf, 1998; Jansen, Ballintijn, Brinkkemper & van Nieuwland, 2006). Every customer has to maintain their own deployment, including the provision of hardware, hosting, deployment, and configuration updating. It also means that, a customer physically owns and controls the data. This implies that customers have more control on the management and access to their data.

In the on-premises situation, a software vendor incorporates specific requirements of a customer by writing specific code and deploying it on-premises at the customers' places. These product customizations frequently form a part of the revenue model of the software vendor, since developing and maintaining customer specific code can be charged to the customer.

4 FOUR PROCESS CHANGE DOMAINS: BUSINESS PERSPECTIVE

In this section, the four domains that require changes for software vendors in adopting the SaaS model are identified and discussed in a business perspective first. There is a change in value proposition while transitioning to the SaaS business model. SaaS creates a value proposition at three levels: utility, process transformation, and business model innovation (Dean & Saleh, 2009; Sarker, Sarker, &

Sahaym, 2012). At utility level, information technology is delivered to customers in the form of a service over the internet rather than a software package (Cusumano, 2008). The SaaS business model, for example, can decrease costs as a result of reduced investments in infrastructure as well as reduce implementation time (Cusumano, 2008; Ming, Kumar, & Whinston, 2009). At process transformation level, SaaS allows customers to collaborate in business processes more effectively by leveraging the availability and assets of cloud computing (Dean & Saleh, 2009). At business model innovation level, SaaS enables organizations to create new business models by linking, sharing, and combining resources in a business ecosystem (Dean & Saleh, 2009). For instance, whereas previously every customer of a software product had to implement its own customized connections to third-party systems, SaaS vendors can connect their offerings to other services and offer the connection as a feature, enabling tightly coupled service integration. However, these characteristics of SaaS raise more questions: what type of value proposition should SaaS providers create for their customers? What should a SaaS provider do in order to create these value propositions? Considering the following aspects will help researchers and practitioners to answer these questions thus facilitating the transition process from on-premises to SaaS business model.

4.1 Changes in business structure

The SaaS business model requires a different set of resources in order to develop and deliver SaaS based solutions, (e.g., knowledge, software development tools, and outsourcing partners). The SaaS business model impacts existing cost structures because companies incur additional costs related to developing, hosting, supporting, and managing SaaS based solutions (Jaekel & Luhn, 2010; Chong & Carraro, 2006). It is most common to employ direct online sales, marketing and customer support channels. However, depending on the complexity of the solution, different channels and partners could be employed in order to market and sell the software as a service as well as provide customer support (Tyrväinen & Selin, 2011). To successfully implement the SaaS business model, an organization should allocate resources and reorganize its structure to support these changes.

4.2 Changes in revenue logic

With the SaaS business model, revenue is not generated by sale of licenses but rather through sale of services. The most common revenue streams of SaaS vendors are: i) subscription fees charged monthly or annually per user; ii) advertising based revenue; iii) transaction based revenue where customers are charged based on the number of transactions they perform; iv) premium based revenue where revenue is generated by providing the basic version of the application for free and charging for premium versions; and v) revenue based on implementation and maintenance services (Cusumano, 2008; Mathew & Nair, 2010; Walsh, 2009). In traditional pricing strategies, vendors commonly use fixed prices or pay per use (Abdat, 2009; Lehmann and Buxmann, 2009). Within a SaaS solution, however, it is not uncommon to charge customers per feature, instead of the complete service. Software vendors should have a clear understanding of their customers' needs, the cost structures of SaaS, and competitors pricing strategies. Based on this kind of information, companies should accordingly adapt appropriate revenue logic strategies (Harmon, Demirkan, Hefley & Ausekils, 2009).

4.3 Changes in customer relationship

Transitioning to a SaaS business model impacts customer relationships (Chong & Carraro, 2006; Deeter et al., 2008; Tyrväinen & Selin, 2011). Since the deployment method changes, the private data of customers is stored at the vendor, a third party hosting provider, or even at an unknown location 'in the cloud'. Software vendors become responsible for the safe storage of data, something that was no issue in the on-premises situation, because customers were responsible for their own data. A SaaS provider should consider licensing agreements including service level agreements, security, access, and data backups to provide more attractive offers to potential customers and build long-term relationships with them.

4.4 Changes in partnerships

Companies face the dilemma whether to create new channels for SaaS goods or to leverage their existing channels (Wainewright, 2009). Companies are adopting an ecosystem approach or value network approach to co-create and deliver customer benefits. This approach is significantly different from the on-premises business model where partnerships are mainly sought for delivering software and services to customers. However, in the SaaS environment, a SaaS provider should consider new partners. For example, SaaS deployments require great amounts of storage and performance capacity to store and handle all transactions of customers. In this case, it must cooperate with infrastructure providers (e.g., data warehouse provider) (Jaekel & Luhn, 2010), because it is more cost effective and operationally efficient to work with infrastructure providers instead of hosting all necessary infrastructures for their customers. Companies should also consider partnering with value added partners (for more details please see case study, changes in business/product structure section) such as add-on software developers.

5 FOUR PROCESS CHANGE DOMAINS: TECHNOLOGICAL PERSPECTIVE

In the technological perspective, a software product often does not have to be redeveloped all over in order to offer the product as a service through the Internet instead of having to ship it to a physical location. Figure 2b shows a simplified representation of the SaaS deployment method. After a product is developed, it is deployed at a service provider; this can either be a department in-house at the software vendor, or an external partner specialized in product hosting. The costs of deployment and management of an installation is absorbed by the SaaS provider. This delivers significant cost savings for both parties due to scalability, even though the SaaS vendor will include parts of the deployment costs in its service fees.

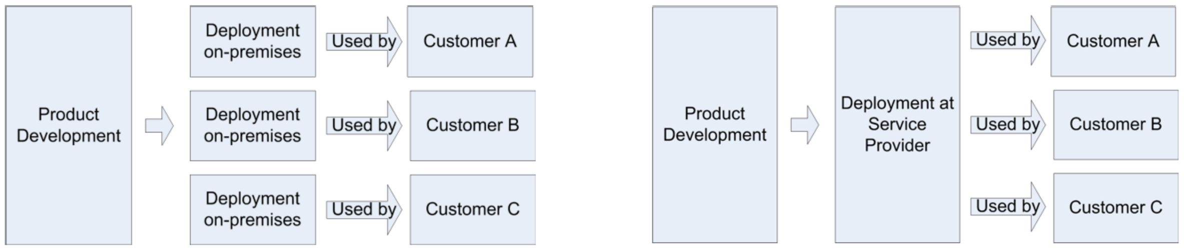

Figure 2a & b. On-premises (a) versus Software-as-a-Service Deployment Method (b)

The ability to serve multiple customers through one shared instance of a software product is called "multi-tenancy" (Dimitrakos et al., 2003). In multi-tenancy, services, both database (Aulbach & Grust, 2008) and software instances (Guo et al., 2007) can be shared by multiple tenants. This means no separate database and software instances have to be maintained or deployed per customer. A possible negative effect of serving all tenants from one shared database or software instance is the reduced flexibility in implementing customers' requirements. This reduction can be counteracted by introducing runtime variability in the SaaS product (Kabbedijk and Jansen, 2011; Mietzner, Metzger et al., 2009). Implementing variability in multi-tenant SaaS environments has many consequences for the software architecture (Zhang et al., 2010; Bezemer & Zaidman, 2010). A large technical advantage of delivering a service through the internet is that updates, which normally require manual work of releasing, delivering, and deploying the update (Jansen et al., 2006), is now simpler since the application is hosed and managed by the software producing organization, possibly reducing time to release innovations within the production version of the product (Carmel, 1995).

5.1 Changes in product structure

The biggest change in software products when moving toward the SaaS model is that multiple different tenants will be using the same instance of the product at the same time. This means that the technical architecture should be changed to manage multiple tenants. This change takes place at database level (Bezemer & Zaidman, 2010), where data of different tenants has to be identified, but also at process level, where different tenants use different functionality of the software product compared to each other. Previously, product security could be left for a large part in the hands of customers. In the SaaS deployment model, however, service providers need to make sure that their customers' data is protected, both from access by unauthorized parties and from access by other customers sharing the same resources.

Releasing new versions of software also changes since software vendors want the least downtime possible, to adhere to service level agreements. In addition, software products are often deployed on multiple servers. In the case of products with a large customer base, releases of new versions, updates, and bug fixes should be rolled out in different phases and not all servers at the same time (Talwar et al., 2005). We define this process *as partial deployment*.

5.2 Changes in revenue logic

Because of the variable nature of multi-tenant systems, the software can be different for many customers. Enabling the pay per functionality option can be difficult for software vendors, because they need a system to monitor the usage and design complex revenue schemes. From a development point of view, it means functionality has to be clearly defined and designed in a way it can function independently. Since customers pay by monthly fee, by transaction and by user, software vendors should develop a sophisticated system to monitor and manage all these different revenue agreements with customers.

5.3 Changes in customer relationship

The shift of data responsibility from the customer to the software vendor leads to the fact that software vendors have to think about different ways to store the data of multiple customers. Customers need a way to be sure their data is safe and the vendor can be trusted. Partnering with a trusted, well-known hosting provider is a way to assure customers of the safety of their potentially sensitive data. Another frequently applied solution is the use of one multi-tenant database to store all tenant data and the possibility for customers to export their data, to give them the feeling of data control. In some cases, like in the case company, customers can choose on what physical database server their data is, but most often customers have no say in this.

5.4 Changes in partnerships

In the on-premises model, third party software developers can offer extensions to customers who purchased software. In the SaaS model, the software vendor has to enable third party software developers to communicate with its running service through an API or other interface. This leads to a situation in which a software company has to take care of the management of all connections with third party developers. It has to make sure that tenants who want to share data with a third party developer can do so; and that tenants who do not want to share certain data should not share it with a third party developer by accident. The structure of the partnership model between the software vendor and third party developers can change extensively because of the changes in software product collaborations (van Angeren et al., 2011).

6 CASE STUDY: THE TRANSITION OF A LARGE ERP VENDOR

Exact is a Dutch ERP software vendor producing both on-premises and SaaS software solutions. This case study focuses on two products, namely ERP-Online and ERP-Onpremises (we named them different from the actual names for clarity). ERP-Online is a SaaS based solution and ERP-Onpremises is an on-premises software product. These two products have been chosen because they provide similar functionality (value proposition) and target the same market segment. Figure 3 shows a high level representation of Exact's on-premises business model and SaaS business model.

6.1 Changes in business/product structure

"One of the biggest challenges and dangers we faced while transitioning from on premise solution provider to a SaaS solution provider was our existing culture. We are predominantly an on premises software developer so we had to make sure that our existing culture should not hamper the new SaaS product. That's why we spun off a separate department who was responsible for ERP-Online" - Manager from Exact

Exact changed their internal structure and spun off a SaaS team who is responsible for developing, supporting and selling ERP-Online. Within this team, one developer has the responsibility to monitor the performance on a daily basis in order to anticipate possible bottle-necks or errors that could compromise the product's uptime.

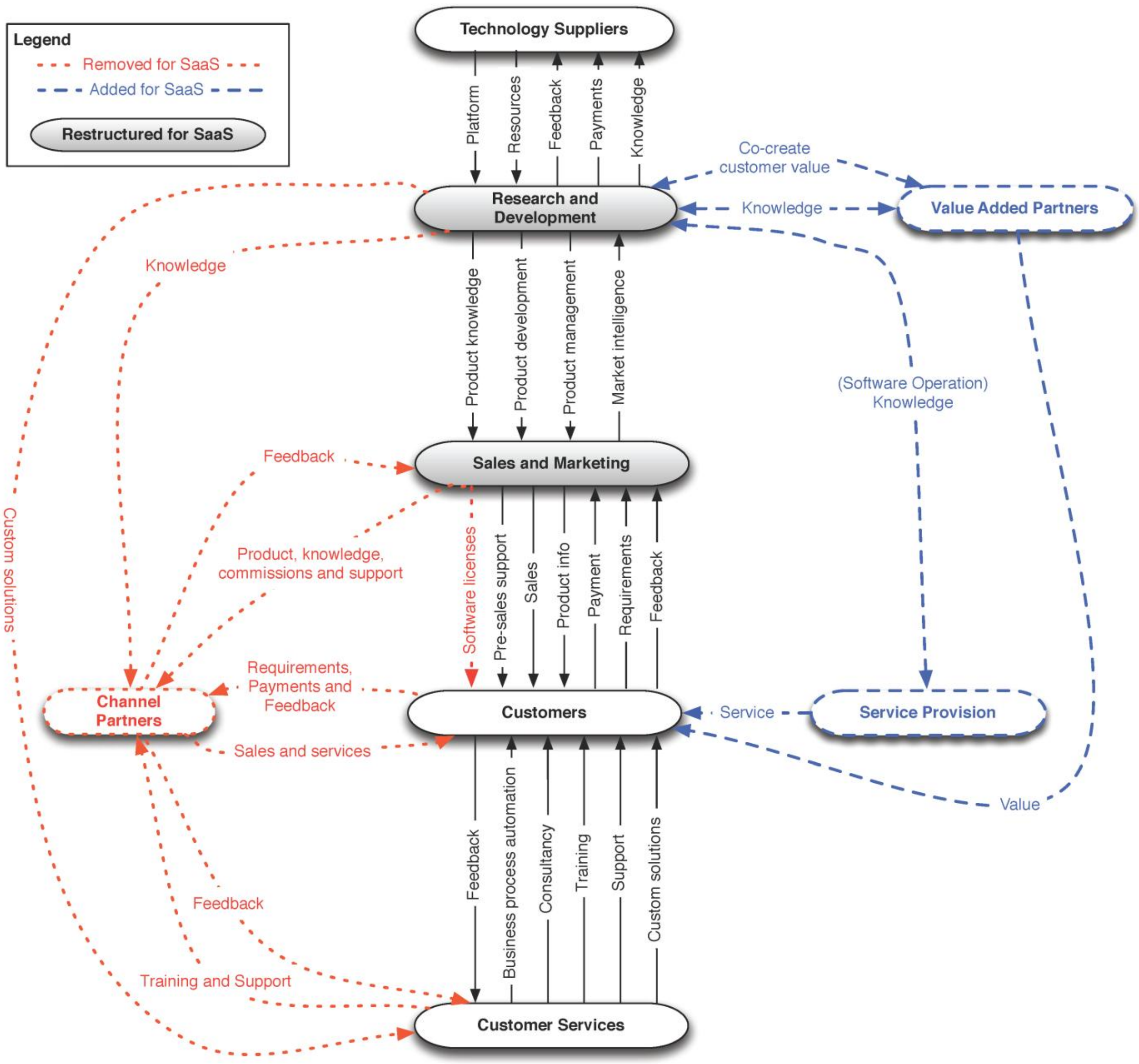

Figure 3. Transitioning from On-premises to SaaS in the Case Study

Looking at Figure 3, it can be observed that Exact makes use of different resources for developing, selling and supporting the software under the two business models. Under the on-premises business model, Exact had to collaborate with Microsoft, and outsourcing partner in case of overflow of software development work. In case of the SaaS business model in addition to Microsoft and outsourcing partner, Exact also deals with a IaaS partner (Infrastructure as a service provider).

It can also be observed that the types of partners that Exact seeks under the two business models are different. Under the on-premises business model Exact focuses on channel partners. In case of the SaaS business model, they focus on establishing and maintaining relationships with value added partners who can co-create value for their customers. Value added partners are partners with whom Exact can build mutually beneficial relationships. For example, ERP-Online has established partnerships with organizations such as banks and SME fuel stations. Under these partnerships, ERP-Online is integrated with information systems of the partnering organizations. This kind of integration allows users of ERP-Online to share relevant information with their accountants by using automated data entry.

"We are looking for partners who can help us co-create value for our customers. We also look at how we can generate revenue from these partnerships" - Manager from Exact

Furthermore, under the SaaS business model, Exact has adopted a direct go to market strategy, i.e., they do not employ their existing distribution channels. This strategy could be attributed to the simplicity of ERP-Online.

From the technological perspective, the architecture of ERP-Online has been changed in several perspectives from the traditional ERP-Onpremises product. Since multiple customers from different companies will use the application at the same time, tenant management becomes very important, especially on database level. This means all records had to be identified by a unique tenant ID. Also changes have to be made to the presentation layer of the product, because the service has to be presented through a browser. Different built up pages have to be presented to all tenant using the product. Another important change that can be observed is the offering of several add-ons that can connect customer's data to third party services. For example, customers can now link their bookkeeping account to their bank account and gather financial information automatically.

6.2 Changes in revenue logic

Looking at Figure 3, it can be observed that there is fundamental shift in revenue logic of Exact. The main sources of revenue under the on-premises business model are sale of software licenses and maintenance fees. However, under the SaaS business model, revenue is being generated through sale of software in the form of a service, or, in other words through sale of subscriptions. For example, Exact charges their customers a monthly fee for the ability to share their banking information with their accounting firms. ERP-Online offers several additional modules for a fee that add functionality to the standard software solution, for example, the support for foreign currencies. Exact expects larger customers to have more specific wishes than their current customer base, and plans on offering smaller modules containing specific functionality. This change can also change their revenue model, since customers will pay based on the functionality they use.

ERP-Onpremises	**ERP-Online**
• Sales of licenses directly as well as through partner network • Maintenance fees • Training & sale of extensions • Consultancy	• Subscription (sale of software in the form of a service) • Consultancy • Upgrades • Recurring transaction based revenue • Revenue from sale of software development kits to partners

Table 1. Comparison of revenue sources between Exact's on-premises and SaaS models

6.3 Changes in customer relationship

The role of Exact has changed from a software licensor to software as a service provider. This has implications for the customer relationship that Exact establishes and maintains with its customers. First of all, an analysis of the relationships established under the two business models reveals that Exact has tried to establish automated relationships with their customers wherever possible. For example, under the SaaS business model, it is possible for customers to purchase and use the application without interacting with anyone within Exact, this is not done under their on-premises business model. Customers have to interact with the sales team in order to purchase a license for ERP-Onpremises.

There is also a change in the licensing agreements that govern the relationships with customers from a software licensor to a service provider. Since Exact now (through their hosting provider) hosts the data of all customers, it is important to have clear agreements on this topic. Customers want to know what their hosting providers use and how their data is handled (security, backups etc.) and who has access to their data. Although most customers are not aware their data is stored elsewhere, it is important to achieve a high level of trust from customers. Exact also has clear agreements on service level and security.

6.4 Changes in customer partnerships

The type of partners Exact pursues under their two business models differs. In the SaaS business model, Exact had to bring onboard a critical partner who provides them with infrastructure as a service. Under the on-premises business model, Exact uses a direct go to market strategies as well as they employ their channel partners to deliver ERP-Onpremises. In case of ERP-Online, channel partners are not employed. The reason for not employing channel partners could be attributed to the simplicity of the solution. This is confirmed by the following quote.

"We are planning to release larger and more complex solutions, which would require more services, and in this case we are planning to involve our channel network. We are currently discussing with our partners on how we could best work together on delivering and supporting SaaS solutions" - Manager from Exact

An overview of partner types per product cn be found in table 2.

ERP-Onpremises	**ERP-Online**
• Vendors (Microsoft, outsourcing partners) • Channel partners	• Vendors (Microsoft, IaaS, outsourcing partners) • Value added partners (e.g. banks) • Partners who develop add-ons for ERP-Online • Partners who help increase the penetration ERP-Online

Table 2. Comparison of partners under the two business models

7 ON-PREMISES TO SAAS TRANSITION MODEL

Based on the findings from literature and the case study, the On-Premises to SaaS Transition Model is created. Table 3 presents the transition model.

		On-premises	**SaaS**
Business/ Product Structure	**B**	Main costs related to: • Research and development • Sales and marketing • Customer support	In addition to costs mentioned under the on-premises model, there are hosting costs.
	T	After releasing, product is shipped to the customer	• Active tenant management • Tenant-aware database • Extensive logging system
Revenue Logic	**B**	Sales of licences	Sales of services
	T	Pay per user	• Pay per use • Pay per feature
Customer Relationship	**B**	Governed by contracts that mainly deal with warrants, customer supports, and upgrades	Shifts towards outsourcing type of contract where additional elements such as service level agreements and security are covered
	T	Customer hosts and owns all data	Software vendor is responsible for data security and safety
Partnerships	**B**	Main focus is on channel partners	Ecosystem approach
	T	Third party connection on premises	• Third party service connection is taken care of by software vendor • Easy connection for third parties through application program interfaces

Table 3. On-Premises to SaaS Transition Model (B: business perspective; and T: technological perspective)

8 DISCUSSION AND CONCLUSION

This paper provides a comprehensive overview of changes that need to be made by a software producing organization. The presented process change domains from both a business and technological perspectives are based on an extensive case study. The novelty of this work lies mainly in the integration of the two perspectives, the framework for assessing the impact of a SaaS transition, and its application to the case of the large software vendor. An overview of all changes can be found in Table 3.

The main strength of this model is also its weakness. It is high-level and comprehensive and thereby should fit any appropriate software producing organization that is transitioning to SaaS. Its comprehensiveness however, forces the model to oversimplify highly complex aspects. The extensive case study makes it possible to evaluate the model, but not necessarily validate completely. This is left to future work.

Further research should give more insights on how software vendors can use this knowledge to make a smooth transition from a on-premises model to a SaaS model, without making common mistakes. Furthermore, many challenges lie in the architectural facilitation of multi-tenancy in SaaS applications that can enable similar customization freedom as traditional on-premises applications, while keeping the advantages of scalability. In the future, we plan to apply the framework to more cases to enrich it with more detailed changes.

References

Abdat N., Spruit, M., Bos, M. (2010). Software as a Service and the Pricing Strategy for Vendors. In *Digital Product Management, Technology and Practice: Interdisciplinary Perspectives*, 154-192.

Albach, S., Grust, T., Jacobs, D., Kemper, A., Rittinger, J. (2008). Multi-tenant databases for software as a service: schema-mapping techniques. In *Proceedings of the ACM SIGMOD international conference on Management of data*, p. 1195-1206, ACM, New York, NY, USA.

Angeren, van J., Kabbedijk, J., Jansen, S., Popp, K. M. (2011). A Survey of Associate Models used within Large Software Ecosystems. *In Proc. of the 3rd Int. Work. on Software Ecosystems*. 1-13.

Bezemer, C., Zaidman, A. (2010). Multi-tenant SaaS applications: maintenance dream or nightmare? In *Proc. of the Int. Workshop on Principles of Software Evolution*, p. 88-92, ACM, New York.

Carmel, E., 1995. Cycle time in packaged software firms. *Journal of Product Innovation Management* 12 (2), 110–123.

Carzaniga, A., Fuggetta, A., Hall, R. S., Heimbigner, D., van der Hoek, A., Wolf, A. L. (1998). A Characterization Framework for Software Deployment. *Technical Report* CU-CS-857-98 University of Colorado.

Chong, F. and Carraro,G. (2006). *Architecture strategies for catching the long tail*. http://msdn.microsoft.com/en-us/architecture/aa479069. accessed. 13-June-2011.

Chou, T. 2005. *The end of software: Transforming your business for the on demand future*. Sams publishing.

Cowley, S. and McLaughlin, K. (2007). *SaaS: It's here to stay*. CRN, Issue 1245

Cusumano, A.M. (2008). The changing software business: moving from products to services. *Computer*, 41(1).

Dean, D. & Saleh,T. (2009).Capturing the value of cloud computing. *The Boston Consulting Group*. http://www.bcg.com/documents/file34246.pdf

Deeter, B., Cowan, D., Goodman, B., Botteri, P., Levine,J., Sarin,A.,Garg,D., Fisher,A. and Messiana,G. 2008.Bessemer's top 10 laws for being "SaaS-y". Bessemers venture partners.

Delattre, M., Ocler, R., Moulette, P. and Rymeyko, K. (2009). Singularity of qualitative research: From collecting information to producing results. *Journal of Critical Postmodern Organization Science*, 7(3), 33-50.

Dimitrakos, T., Mac Randal, D., Yuan, F., Gaetta, M., Laria, G., Ritrovato, P., Serhan, B., Wesner, S., Wulf, K. (200t3). An Emerging Architecture Enabling Grid Based Application Service Provision. In *Proc. of the 7 h Int. Conf. on Enterprise Distributed Object Computing*, p. 240-251, IEEE Press.

Guo, C., Sun, W., Huang, Y., Wang Z. H., Gao, B. (2007). A Frameworkthfor native Multi-tenancy Application Development and Management. In *Proceedings of the 9 IEEE International Conference on E-Commerce Technology*, 551-558, IEEE Press.

Harmon, R., Demirkan, H., Hefley, B. and Ausekils, N. (2009). Pricing strategies for information technology services: A value-based approach. In *Proceedings of the 42nd Hawaii International Conference on System Sciences.*

Jaekel, M., and Luhn, A. (2010). *Cloud computing - business models, value creation dynamics and advantages for customers - White paper*. Siemens IT Solutions and Services GMBH.

Jansen, S., Ballintijn, G., Brinkkemper, S., van Nieuwland, A. (2006). Integrated Development and Maintenance for the Release, Deployment and Customization of Product Software: a Case Study in Mass-market ERP Software. *Journal of Software Maintenance and Evolution: Research and Practice*, 133-151, 18.

Kabbedijk, J., Jansen, S. (2011). Variability in Multi-tenant Environments: Architectural Design Patterns from Industry. *Lecture Notes in Computer Science*, 151-160, 6999.

Kaplan, M.J. 2005. SaaS Survey Shows New Model Becoming Mainstream. *Cutter Consortium Executive Update*, 6(22), 1-5.

Kontio, J., Jokinen, J., Mäkelä, N.M. and Leino, V. (2005). Current practices and research opportunities in software business models. *ACM SIGSOFT Software Engineering Notes*, 30(4).

Krasner, G. E., Pope, S. T. (1988). A Cookbook for using the Model-view Controller User Interface Paradigm in Smalltalk-80. *Journal of Object-Oriented Programming*, 1(3), 26-49.

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

Manuel, P. D., AlGhamdi, J. (2003). A data-centric design for n-tier architecture. *Information Sciences*, 150(3), 195-206.

Mathew,M. & Nair,S. 2010. Pricing SaaS models: Perceptions of business service providers and clients. *Journal of Services Research*, 10(1), 51-68.

Mietzner, R., Metzger, A., Leymann, F., Pohl, K. (2009). Variability Modeling to Support Customization and Deployment of Multi-tenant-aware Software-as-a-Service applications. In *Proc. of the ICSE Workshop on Principles of Engineering Service Oriented Systems*, 18-25.

Ming, F., Kumar, S. & Whinston, A. (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.

Mousavidin, E. & Silva, L. (2009). Packaged software configuration through the lens of social construction of technology. In *Proc. Of the 42nd Hawaii Int. Conf. on System Sciences*, 1-8.

Popp, K.M. (2011).Software industry business models. *IEEE Software*, 28(4), 26-30.

Rajala R, Rossi M, Tuunainen VK (2003a) A framework for analyzing software business models. In *Proceedings of the 11th European conference on information systems.* Naples, Italy

Rauscher, G. & Smith, P.G. (1995). From experience time-driven development of software in manufactured goods. *Journal of Product Innovation Management*, 12 (3), 186-199.

Sarker, S., Sarker, S. & Sahaym, A. (2012) Exploring Value Cocreation in Relationships between an ERP Vendor and Its Partners: A Revelatory Case Study. *MIS Quarterly*, 36(1), 317-338.

Stamford, C. (2011). Gartner says worldwide software as a service revenue is forecast to grow 21 percent in 2011. http://www.gartner.com/it/page.jsp?id=1739214 accessed on 10-July- 2011.

Talwar, V., Milojicic, D., Wu, Q., Pu, C. & Jung, G. (2005). Approaches for Service Deployment. *IEEE Internet Computing*, 9(2), 70-80.

Tyrväinen, P. & Selin, J. (2011). How to sell SaaS: A model for main factors of marketing and selling Software-as-a-Service. *The 2nd International Conference on Software Business.*

T Valtakoski, A. and Rönkkö, M. (2010). Diversity of Business Models in Software Industry, pp 1-12. Proceedings of the International Conference on Software Business. Lecture Notes in Business Information Processing, Springer Berlin Heidelberg.

Walsh,B. (2009). *The web startup success guide*. Springer - Verlag New York Inc.

Wainewright, P. (2009). Debunking myths about the SaaS partner channel. http://goo.gl/zxqQ5 accessed: 10-Fri-2011.

Yin, R. (2009). *Case Study Research: Design and Methods*. 4th Edition. Sage Publications, California.

Zhang, X., Shen, B., Tang, X., Chen, W. (2010). From isolated tenancy hosted application to multi-tenancy: Toward a systematic migration method for web application. In *Proc. of the Int. Conf. on Software Engineering and Service Sciences,* p. 209-212, IEEE Press.