Towards Requirements Engineering for "Software as a Service"*

Marina Berkovich[1]*, Sebastian Esch*[1]*, Jan Marco Leimeister*[2]*, Helmut Krcmar*[1]

[1]*Lehrstuhl für Wirtschaftsinformatik*
Technische Universität München
Boltzmannstr.3
85748 Garching

[2] *Fachgebiet Wirtschaftsinformatik*
Universität Kassel
Nora-Platiel-Str. 4
34127 Kassel

1 Introduction

More and more companies realign their competencies to offer integrated bundles of products and services with the goal to solve customer problems (Leimeister et al. 2009). The customers are not interested in products or services per se, but they want their problems to be solved (Böhmann und Krcmar 2007). Offering a solution has advantages for the customer and the supplier. The customer gets an individualized solution for his problem and the supplier reaches higher profits in comparison to a "classic" product (Galbraith 2002).

This trend from offering traditional products to offering solutions can be observed in product development, known as Hybrid Products or Product Service Systems (PSS), as well as in the software industry, where this trend is called Software as a Service (Böhmann und Krcmar 2007; Berkovich et al. 2009a).

The success of a product depends on how good it satisfies the requirements and wishes of the customer. To assure this success, it is important to carry out requirements engineering accurately (Lindemann 2006). "Requirements engineering (RE) — the elicitation, definition, and management of requirements — is often cited as one of the most important, but difficult phases of software development (Damian und Chisan 2006)." Faulty RE is one of the main reasons for the failure

* Dieser Beitrag ist im Rahmen des Sonderforschungsbereiches 768 „Zyklenmanagement von Innovationsprozessen – Verzahnte Entwicklung von Leistungsbündeln auf Basis technischer Produkte", gefördert von der Deutschen Forschungsgemeinschaft (DFG), sowie des Forschungsprojekts SPRINT entstanden, gefördert durch das Bundesministerium für Bildung und Forschung unter dem Förderungskennzeichen 01FD0609.

of development projects (Aurum und Wohlin 2005). The later errors are detected in the development process, the higher are the costs resulting from them. Companies are forced to develop more complex and more innovative systems and concurrently adapt them to the requirements of the customer (Pohl 2007). Especially interesting is how RE for SaaS as a hybrid product has to be done. This paper discusses the requirements for the requirements engineering process of SaaS.

2 Software as a Service

Software as a Service (SaaS) is defined as a software offered as a service that can be accessed via the Internet (Buxmann et al. 2008). SaaS is an important trend in the IT domain as can be seen in the increasing market share of about 50% per year (Choudhary 2007). A main characteristic of SaaS is that it separates the owner and the user of software. Another characteristic of SaaS is that it allows offering of product-variants, depending on the needs of the customer (Choudhary 2007).

Often SaaS is mentioned in conjunction with ASP (Application Service Providing). ASPs offer COTS (components-off-the-shelf) software hosting and support, services supporting multiple users' access to an application that is centrally managed via a subscription-based access model (Liang et al. 2006). In distinction to ASP, software offered as SaaS supports multiple customers sharing the same software instance, different configurations for distinct customers or recording of usage information for new business models. SaaS is also mentioned in conjunction with Service-oriented Architectures (SOA). SOA is a technical means to integrate different applications communicating via a network, often used by SaaS (Buxmann et al. 2008). In contrast to SOA, SaaS is not concerned with the technical realization, but with the appearance of the services themselves. A similar concept to SaaS is that of offering solutions consisting of customized versions of MOTS (modifiable off-the-shelf) software defined by Carney and Leng (2000). These solutions consist of software and consulting activities aiming at integrating software into the business processes of the customer (Böhmann und Krcmar 2007), by modifying the software itself. In comparison to SaaS, customized MOTS has no service component during the software's usage phase. As the customer-individualization is a main characteristic of SaaS it is often handled in conjunction with MOTS. SaaS offers software like MOTS or COTS, with the distinction that during the development of SaaS, the business case "as a service" was already incorporated.

SaaS can be seen as a complex solution, aiming at solving customers' problems and providing advantages for the customer. MOTS is offered as a service in such a complex solution. From another point of view, SaaS can be a part of a complex solution, supporting face-to-face services. Both perspectives lead to different aspects that need to be regarded when developing the software. Such a complex solution is also known as a hybrid product (or PSS - product service system) (Leimeister und Glauner 2008). A hybrid product is a combination of physical products and services offered on the market as integrated service bundles (Becker und Krcmar 2008). In this context software is also regarded as a product, since it plays a substantial role in modern products.

Hybrid products can be characterized by the degree of individualization, process-integration and service-integration (Sawhney 2006). Service-integration means coordinated integration of product and service. Thereby the benefits for the customer in purchasing service bundles are higher than if he purchases the single product- and service-components separately. Process-integration means that hybrid products are seamlessly integrated into the processes of the customer. Therefore the supplier of hybrid products has to thoroughly know the customer processes. Individualization means that hybrid products have to be adapted to the customer wishes (Sawhney 2006). Becker and Krcmar (2008) point out that a bundle of the components, without any added value for the customer, does not increase the willingness to pay but does instead increase the desire for discounts.

Thus, the development of hybrid products requires a high degree of technological integration reflected by the integration of technological and organizational parts, the individualization, which means that the hybrid products are customized, and the transformation of solutions, which means that the customer-value is increased by the solution, and the different lifecycles of hybrid products' components (Böhmann und Krcmar 2007; Berkovich et al. 2009b). These aspects have the consequence that the development of hybrid products has to be done differently than that of common products. In literature as well as in practice the parts of a hybrid product are handled separately (Leimeister und Glauner 2008).

There are also SaaS that do not fulfill the definition of hybrid products. An example for such a SaaS is the Google calendar. It is offered through internet and can be used by many different people. But it does not fulfill the criteria of individualization. Every user does use exactly the same software without the possibility to request individual changes from the solution provider.

In the focus of this paper are only SaaS that are also hybrid products. Therefore they regard the customer-individualization as important. That means the solution or parts of the solution have to be adapted to accurately fulfill the customer needs. "Software is developed to meet necessary and sufficient requirements, in a highly personalized, fine-grained, and transparent manner (Chen et al. 2005)".

3 SaaS in the context of Hybrid Products

To analyze the requirements for the RE for SaaS as a hybrid product, it is necessary to understand the character of SaaS. Most papers discussing SaaS show different fields of application, general aspects regarding business economics, different views on the software- or service-part or they describe technical aspects.

In the context of software as a service and hybrid products two scenarios arise merging both topics, leading to new challenges for RE. In the first scenario SaaS itself can be seen as a hybrid product (Figure 1), a bundle of software and services to address the needs of the customers. The software is often especially developed to be offered as a service, considering different requirements concerning data-handling or supporting different customers on the same software instance.

An example for SaaS is a CRM-software provided as a service through the internet (http://www.salesforce.com) that offers a web-based solution for sales, service, marketing and call-center. It can be characterized as a hybrid product.

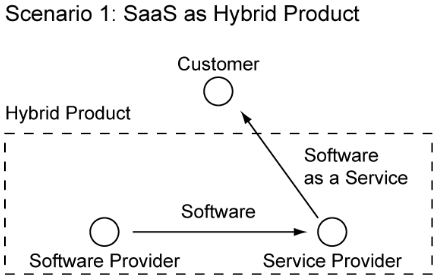

Figure 1: SaaS as hybrid product (Source: Own illustration)

Customer and service provider are different stakeholders, with different characteristics and requirements. Customers have requirements for the whole SaaS product, including service aspects and software functionality. They can be seen as a market, where market analysis techniques are necessary to elicit requirements. Service providers have requirements to integrate the software into their service processes. So the development of tightly integrated software and services requires close cooperation between software provider and service provider, but the service processes focus on deploying, delivering and supporting the software. In this scenario as well as in the second scenario, software/technology provider and service provider can be organizational units of the same organization or different organizations (Becker und Krcmar 2008). If both stakeholders are from the same organization, boundaries during the development process are easier to overcome.

In the second scenario, software as a service is used to enable face-to-face services (Figure 2). An example of such a hybrid product is a sports-coaching-service for workplace-health-promotion. Thus the participants in such an activity program use the software like standard software to view their workout schedules, document their activity and communicate with their coaches. The coaches of the service provider use the software to monitor and manage the participants. Through the use of the software, they support more participants as without support through software.

In this scenario, the functional requirements are not only derived from the customer but also from the service provider, who needs specific functionality to support his service processes. In contrast to the first scenario the coupling between service processes and software functionality is tighter. The requirements regarding deploying, delivering and supporting the software shift from the service provider in the first scenario to the technology and service provider. Also needs arise to decide together, which service processes can be automated or supported through the software, to make the hybrid product efficient and scalable.

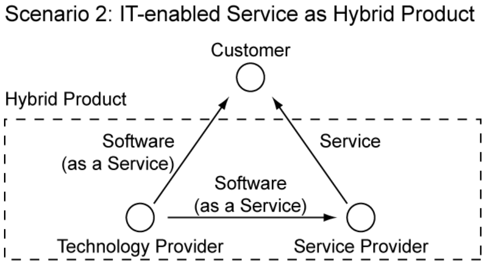

Figure 2: IT-enabled service as hybrid product (Source: Own illustration)

In both scenarios it is necessary to manage requirements for the hybrid product as a whole and their mappings to requirements for specific service- and software-components. The different stakeholders involved in developing the hybrid product have different requirements: Customers have requirements on the hybrid product as a whole. They do not differentiate between software and service components. Customers have requirements regarding functionality, availability (software and support), the integration into their own processes (key characteristic of hybrid products) and the security of their data. Service providers have requirements on software and service components. They may have a usage-based business model and corresponding requirements; e.g. that the software has to support usage-tracking. Other requirements may regard easy provisioning of the software, functionality to ease customer support or integration into customer processes.

In the second scenario, there are two SaaS relationships: between technology provider and customer and between technology provider and service provider. In the first case, the software offered is standardized for all customers with only little variance. In the second case the need for modularization and customization is stronger, especially if there is not only one service provider but different service providers, that use the same software to provide the service, for example in different geographical regions.

4 Requirements Engineering

As described in earlier sections SaaS is characterized by customer-individualization. Therefore the development of such products requires tight involvement of the customer. Only this way it can be assured that the needs of the customer are well understood and realized adequately. The integration of the customer is especially relevant for requirements engineering. The RE has the task to collect all requirements regardless of their source. The requirements have to be managed for the solution as a whole. Requirements to different parts of the solution have to be stated commonly to ease interdisciplinary communication. Nonetheless differences between the different domains and their components have to be regarded; i.e. different requirement elicitation methods are suited for different domains.

The first phase of a development project is to gather the requirements. Within the next phases of the development it is necessary to manage the requirements. In software engineering requirements engineering is seen as a special field that is "concerned with the real-world goals for functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families. The subject of requirements engineering is inherently broad, interdisciplinary, and open-ended" (Zave 1997). Berenbach (2006) points out that the increasing complexity of products and projects results in the growing importance of RE in the development process. Defining the requirements for RE for hybrid products requires to understand the concepts of RE.

According to Sommerville and Kotonya (1998) RE consists of the following phases: elicitation, analysis and negotiation, documentation, and validation. Within the requirements elicitation the goals, wishes and requirements from the different stakeholders are to be gathered. In the following phase the requirements are modeled and documented. During the analysis the initially elicited requirements are refined and concretized. The purpose of requirements negotiation is to resolve conflicts regarding requirements between stakeholders. Validation of requirements is defined as „the process of evaluating software at the end of the software development process to ensure compliance with software requirements“ (IEEE 1990).

Requirements traceability "refers to the ability to describe and follow the life of a requirement, in both, forwards and backwards, directions" (Gotel und Finkelstein 1994). In connection with the traceability it is important to assess the impact of changes of requirements to the already developed system and to other requirements. These assessments are being done during the change management.

5 Requirements for RE for SaaS

Up to now there exists no dedicated RE for hybrid products. In literature all involved domains have their own RE models, but there is no integrated one. SaaS consists of immaterial parts as software and services but also material parts as hardware on which the software is running. As a result of that also the lifecycles and the development methods are much different.

In section three it was described that the development of SaaS involves the software provider, service provider and the customer. Regarding the requirements engineering this means that there are two primary stakeholders that propose requirements to the developed solution. The customer has requirements to the software he will use and to the service that he will purchase. The service provider has requirements to the software, because he has to administer the software and he has to provide it to customers. In the second scenario, the technology provider has similar requirements regarding provisioning and maintenance of the software as the service provider in the first scenario. These requirements are very different from their nature. Some are referring to a software product while others are referring to a service. Furthermore these requirements are realized by different domains. These are the main reasons for the challenges for requirements engineering of SaaS.

Based on our review of SaaS as a hybrid product and of RE in general, we derived seven requirements for the RE of hybrid products (table 1). As a justification of these requirements, characteristics of SaaS that motivate the requirements are given. Each requirement is described in detail in the following sub sections.

Table 1: Requirements and their justifications

Characteristic of SaaS as hybrid Products	**Requirement for the RE-Process**
• Integration of multiple components developed by different domains	1) Coordinated and integrated RE for all single components of SaaS
• Interdisciplinary development • Customer and user are not the same	2) Selection of the right stakeholders
• optimal satisfaction of customer needs • customer-individualized products	3) Better customer integration into the RE-process
• different lifecycles of different components of hybrid products	4) Consideration of changes of requirements after/during delivery
• different sources of requirements • single requirements are realized by different domains	5) Thorough and continuous documentation of requirements
• different elicitation methods for different sources of requirements • main stakeholders: service-provider, customer	6) Consideration of the source of requirements during elicitation
• changes can affect both service provider and software developer	7) Consideration of the source of requirements during change-management

5.1 Coordinated and integrated RE for all single components of SaaS

SaaS contains multiple components, such as software and services, which are combined into bundles. These single components, provided by different domains, namely software engineering and service engineering, should be integrated carefully. The single components of hybrid products are actually handled separately by literature and practice (Leimeister und Glauner 2008).

The requirements are usually stated for the entire solution. These requirements have to be split and stated for each part of the hybrid product. The RE-process has to be adjusted accordingly. If the requirements of the entire solution change, the impact of this change on the single parts of the hybrid product has to be evaluated.

5.2 Selection of the right stakeholders

"Requirements come from people, not books" (Alexander 2006). This statement underlines that it is very important to find the "right" stakeholders to get complete information on the requirements. The stakeholders have the task to state the requirements for the solution under development. A stakeholder is defined as "any group or individual who can affect or is affected by the achievement of the organization's objectives" (McManus 2004). A wrong selection of stakeholders, that means a selection of stakeholders that are not relevant for the development of the system, can affect the further development negatively. In section 4 the characteristics and the main stakeholders of SaaS were described. A main stakeholder is the customer that wants to purchase the SaaS. He states requirements on the software and the service. The service-provider itself is an important stakeholder, because he maintains the software and has therefore his own requirements to it. Based on those requirements the software provider has to implement the software. The aspect of selecting the right stakeholders is especially important and difficult in interdisciplinary development. Stakeholders come from different domains and their requirements have to be understood and managed accordingly. In the case of SaaS requirements both on software and services have to be managed.

5.3 Better customer-integration into the RE-process

One of the most important attributes of hybrid products is their orientation on customers' wishes and their satisfaction. The customer acceptance is growing with a higher fulfillment of customer requirements. Most companies have not understood that the integration of the customer into the RE-process helps to increase the understanding of customer-requirements and helps to implement the requirements correctly. Often the customer is only involved in the first phase during elicitation of requirements. Newer approaches from software engineering, especially agile methods like Extreme Programming (Beck 2000) involve the customer during the complete development process. But they don't cover the service engineering part of the SaaS solution. After that phase the customer is hardly integrated into the process until the validation phase. Also in the phase of requirements analysis the customer has to be integrated. This especially facilitates the requirements negotiation, where conflicts between requirements are resolved and where misunderstandings between the customer and the developers are dissolved. The communication with the customer in these phases is problematic, because he has another "language" and another problem-understanding than the developer (Abramovici und Schulte 2007). A precondition for the integration of the customer is that the customer and the developer are able to understand each other. It is important that the customer is not underestimated. The knowledge of the customer about technologies and possible solutions is becoming more and more realistic. The service provider should also be integrated into the RE of the software components, to assure a tight integration of software and service components.

5.4 Consideration of changes of requirements after delivery of the software

After the software development is completed and the software has been delivered to the service-provider, the life of SaaS is not over. Services are continually delivered. Therefore the lifecycles of different components of hybrid products are different. The RE-process has to be able to manage new requirements arising during use of the hybrid product. During the use of SaaS new requirements can be expressed by the customer. In this case it is necessary to analyze whether the software or the service is affected by the change. To do such an impact analysis traceability between requirements and the solution is needed.

5.5 Thorough and continuous documentation of requirements

Because a hybrid product consists of several components developed by different domains, the requirements have also different sources. The requirements on software and services are handled differently. When documenting the requirements its source has to be explicitly regarded, to support the traceability, validation, and handling of requirements after the development.

5.6 Consideration of the source of requirements during elicitation

Due to the insights gained above about the composition of SaaS as hybrid product, the sources of requirements have to be regarded carefully. Requirements for the hybrid product derive from the market. Therefore for eliciting requirements market analysis methods are used. The market analysis has to be carried out both by the service provider and the software producer. Requirements on the software either derive from requirements of the customer of the hybrid product, elicited by market analysis, or from the service provider. The latter can be elicited using methods known from software engineering like workshops, interviews, observations and scenarios (Pohl 2007). The idea of SaaS as a hybrid product is that the software producer develops individual software for a service provider, which himself offers this software as a service to a customer. In this context it is important to shape the communication between software producer and service provider so that it is guaranteed that the requirements of both the market and service provider are considered adequately. The attributes of software in the context of SaaS offered as a software product and as a service should be regarded during the traceability, validation and change-management of requirements.

5.7 Consideration of the source of requirements during change-management

The requirements of both software and service can change. In the case of SaaS as a hybrid product all changes have to be carefully handled. If a change of the Software as a Service is requested, it has to be decided if only the service has to be changed or if the software itself has to be changed. Changes of the service alone have to be done by the service provider, while changes of the software have to be handled by the software producer and the service provider together.

6 Conclusion

In this paper the nature of Software as a Service has been described. SaaS has been set into relation with hybrid products, which have several commonalities with SaaS. The point of view of hybrid products enabled us to take a holistic view on SaaS, without favouring one part of them.

Then the role of requirements engineering in the development of SaaS has been clarified. From the characteristics and needs of SaaS, requirements to the RE process were derived. These requirements describe seven areas of requirements engineering, which are particularly important for SaaS. These areas for improvement should be addressed first by research and practice.

SaaS generates its main benefit by addressing the customer wishes better than classic software does. Thus, one of the main goals is to improve the customer-integration into the RE process. This way experiences with customer-integration, already made in the context of hybrid products can be transferred to SaaS.

The next step in research is to define a requirements engineering model for SaaS, based on the requirements presented in this paper. Such a model consists of an artifact and a process model. The artifact model describes the types of requirements that have to be specified for the solution as a whole and the refined requirements to the single domains involved in creating the solution. Also the relations between requirements beyond domain boundaries have to be captured.

This work will lead to a deeper understanding of requirements engineering and its integration into the overall development process of SaaS. For research this enables the possibility of a systematic integration of existing RE methods and techniques in the development of SaaS. In practice, through a comprehensive RE model, the collaboration between different domains involved in the development is enhanced. For the interdisciplinary communication, a common understanding, like provided by the RE model, is essential. The developed RE model also supports the refinement of requirements alongside the phases of RE and the partitioning of them into requirements for each domain. This way traceability as well as defined responsibilities for requirements can be established in practice. In general, the developed RE model will contribute to a more effective RE in practice and it will therefore improve the overall development process of SaaS.

Literature

Abramovici M, Schulte S (2007) Optimising customer satisfaction by integrating the customer's voice into product development. International Conference on Engineering Design, ICED'07. Paris, France.

Alexander I (2006) 10 small steps to better requirements. In: Software, IEEE, 23(2): 19-21.

Aurum A, Wohlin C (2005) Engineering and Managing Software Requirements (1 Aufl.), Springer, Berlin.

Beck, K (2000) Extreme Programming Explained: Embrace Change, Addison-Wesley, Reading, MA.

Becker J, Krcmar H (2008) Integration von Produktion und Dienstleistung – Hybride Wertschöpfung. In: Wirtschaftsinformatik, 50(3).

Berenbach B (2006) Introduction to Product Line Requirements Engineering Paper presented at the 10th International Software Product Line Conference.

Berkovich M, Esch S, Leimeister JM, Krcmar H (2009a) Requirements engineering for hybrid products as bundles of hardware, software and service elements – a literature review. 9. Internationale Tagung Wirtschaftsinformatik. Wien, Österreich.

Berkovichsp M, Leimeister JM, Krcmar H (2009b) An empirical exploration of requirements engineering for hybrid products. Proceedings of the XVIIth European Conference on Information Systems (ECIS). Verona, Italy.

Böhmann T, Krcmar H (2007) Hybride Produkte: Merkmale und Herausforderungen. In: Wertschöpfungsprozesse bei Dienstleistungen: Forum Dienstleistungsmanagement. Hrsg.: Bruhn, M.; Stauss, B. Gabler, p. 240-255.

Buxmann P, Lehmann S, Hess T (2008) Software as a Service. In: Wirtschaftsinformatik, 50(6):500-503.

Carney D, Leng F (2000) What do you mean by COTS? Finally, a useful answer. In: Software, IEEE, 17(2):83-86.

Chen JC, Gold NE, Mehandjiev N, Layzell PJ (2005) Managing supply chains of software as a service through agent negotiations. Paper presented at the Seventh IEEE International Conference on E-Commerce Technology. CEC 2005, p. 378 - 381.

Choudhary V (2007) Software as a Service: Implications for Investment in Software Development. Paper presented at the 40th Annual Hawaii

International Conference on System Sciences, HICSS 2007, Waikoloa, HI, p. 209a - 209a.

Damian D, Chisan J (2006) An Empirical Study of the Complex Relationships between Requirements Engineering Processes and Other Processes that Lead to Payoffs in Productivity, Quality, and Risk Management. In: IEEE Transactions on Software Engineering, 32(7):433 - 453.

Galbraith JR (2002) Organizing to Deliver Solutions. In: Organizational Dynamics, 31(2):194-207.

Gotel OCZ, Finkelstein CW (1994) An analysis of the requirements traceability problem. Requirements Engineering, 1994., Proceedings of the First International Conference on. Colorado Springs, CO.

IEEE (1990) IEEE standard glossary of software engineering terminology (IEEE 610.12-1990). New York.

Leimeister JM, Glauner C (2008) Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik. In: Wirtschaftsinformatik, 3.

Leimeister JM, Knebel U, Krcmar H (2009) Hybrid Value Creation in the Sports Industry - the Case of the Mobile Sports Companion. In: International Journal of Information Systems in the Service Sector (IJISSS).

Liang D, Xianghua L, Lihua H. (2006) Exploring the new ASP Business Model: Business Circles Oriented ASP Platform. Paper presented at the Service Systems and Service Management, International Conference on, p. 488-494.

Lindemann U (2006) Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden (2 Aufl.), Springer, Berlin.

McManus J (2004) A stakeholder perspective within software engineering projects. Paper presented at the IEEE International Engineering Management Conference, p. 880 - 884.

Pohl K (2007) Requirements Engineering. Grundlagen, Prinzipien, Techniken (1 Aufl.), Dpunkt Verlag.

Sawhney M (2006) Going beyond the Product: Defining, Designing and Devlivering Customer Solutions. In: The Service-dominant Logic of Marketing. Hrsg.: Lusch, R.F. M. E. Sharpe, Armonk, p. 365-380.

Sommerville I, Kotonya G (1998) Requirements Engineering: Processes and Techniques (1 Aufl.), Wiley & Sons.

Zave P (1997) Classification of Research Efforts in Requirements Engineering. In: ACM Computing Surveys, 29(4):315-321.