Design Rules for User-Oriented IT Service Descriptions

Henrik Brocke, Thorsten Hau, Alexander Vogedes, Bernhard Schindlholzer,
Falk Uebernickel, Walter Brenner

University of St.Gallen

{henrik.brocke, thorsten.hau, alexander.vogedes, bernhard.schindlholzer,
falk.uebernickel, walter.brenner}@unisg.ch

Abstract

Customers of complex IT-services increasingly demand integrated value bundles that fit their individual needs. At the same time, IT service providers are facing commoditization of their products and need to standardize their portfolios to realize economies of scale. While approaches to coping with the gap between individual customer demand and the economic necessity of standardization have a long standing tradition in mature manufacturing industries, IT-service providers still struggle with translating their standardized portfolio into a form that is understandable and relevant for their customers. This paper proposes a way of describing IT-services that follows the paradigm of a "service dominant logic". We therefore transfer "service dominant logic" to the realm of IT and propose guidelines to create customer oriented service descriptions. An excerpt of a prototype description serves as an example, how the technical, inside view on IT-services can be translated into a customer-oriented outside view.

1. Introduction

This paper focuses on business-to-business IT-services such as hosting an application or providing desktop services. Providers of that sort of service (IT-service providers henceforth) are increasingly facing challenges of cost pressure and higher quality requirements. This development is largely due to increasing standardization and commoditization of information technology [1, 2]. While competition is thus driving down prices for IT services, customers question the value of IT-services and wonder whether they get their money's worth.

The combination of commoditization of their services and growing customer demands poses a significant threat for the business models of companies offering IT-services [1, 3, 4]. To cope with the cost pressure IT-service providers are facing, they have started to standardize their services. From producing specialized offerings for individual customers, IT-providers have moved to producing standard components at low cost and bundling them into customized service bundles. This approach promises economies of scale in delivering services to customers [5].

While this shift from creating highly individual solutions towards standardized offerings, which can be assembled into service bundles, has been taken up quickly in the industry (customizable standard software such as SAP or blade servers are good examples), a second related evolution has not yet taken place. When markets are shifting from seller markets to highly competitive buyer markets, as is happening in the IT-service market, companies have to focus on the value propositions they are offering to win and retain customers [4, 6, 7]. Other industries that have been confronted with commoditization have learnt to deal with this situation by moving away from a purely product centered perspective towards a service centered perspective, where the tangible product is only one component in a whole bundle. Even manufacturers of such tangible goods as automobiles have turned towards offering "utility" instead of products and today do not simply sell 'cars' but 'mobility' (take GM with services like "GMAC" and "OnStar" for example). The creation of integrated value bundles, that include maintenance, insurance and mobility services together with the base product automobile, is a manifestation of this shift from simple products towards value propositions.

Until today, many IT-service providers have not focused on making value propositions to their customers, but still focus on offering disintegrated commodity products and services. This gives rise to

978-0-7695-3450-3/09 $25.00 © 2009 IEEE

two considerations: Firstly, IT-service providers must focus on making value propositions (in the sense of [8, 9]) to address the customers' needs. Secondly, the value propositions have to be communicated to the customer in an appropriate way, as we will detail below.

Customers therefore should be offered integrated solutions that solve their problems instead of separate products and services, which they have to integrate themselves into usable services. In this article the word service is used in the sense of [8]. Thus a computer screen proposes a value by providing the service of displaying information. Only by looking at it, the customer can realize that value and use it in its process of making value propositions to other entities. Throughout we consider this proposed value to be the relevant aspect of all instances of IT-services. The most relevant distinguishing factor in this work is to whom value is proposed. The IT-service provider is an integrator of different services and bundles them according to the customer's needs. Thus the challenge for the provider is to propose the right service- and value- bundles to the customer to support his/her business process. In the screen example above the right service would be "displaying business data" and not the mere potential to display anything. Thus, one key aspect of the discussion below would be how to identify the customer need and then describe the corresponding integrated service solution in a customer oriented way.

A key aspect for the definition of customer focused services is their description. Nowadays, service descriptions of IT-services exhibit a resource-oriented, inside perspective, since they typically contain an enumeration of all input factors used to create a service and list descriptions and quality parameters of each input factor. Customer-oriented services on the other hand need to focus on the value proposition for the customer. A separation between the inside view and the outside view is necessary. This work does not solve the problem of translating a set of resources into marketable products, since we believe this to be a genuine engineering task. Much rather this work focuses on proposing design rules for developing customer-oriented descriptions.

The remainder of the paper is structured as follows. First we analyze current service descriptions with a focus on SLAs in order to support the thesis, that IT service provider have not yet realized the difference between input resources and outcome value and that current service descriptions are not appropriate means of communicating value to customers. We also analyze some recent research in the realm of SLA design and service descriptions and explain the difference to our work. Then we lay the theoretical foundation for our work by explaining what "service dominant logic" is and by giving a short overview over the concept of "end-to-end" offerings. Following this part, we describe our research process and methodology that led to the rules we created. The main part is dedicated to describing our design rules for customer oriented service descriptions. We finish by drawing some conclusions from our work.

2. State of the Art in Describing Services

This section establishes, that today's service descriptions are not the best way to address customers of IT- services. As shown above, IT services are becoming commodities. Therefore, providers of such services need to focus on creating value propositions and on communicating them attractively to potential customers. Unfortunately this is not the case. Nowadays descriptions are merely a statement of the characteristics of the resources that are involved to provide the service, filled with technical terms and therefore difficult to understand [10]. Due to their binding character, today Service Level Agreements (SLAs) are one of the most crucial description components in business-to-business relationships. But just like the whole service description, their focus commonly is on defining service-level parameters of IT components or input factors and not on the actual end-to-end services that affect the customer, i.e. the outcomes. The definition of service-levels is mostly limited to single components that interact to deliver the IT services. Yet, from the perspective of the customer, the individual service-level agreements are irrelevant, because the service-level is defined through all components that are involved in the service delivery chain [11].

The screening of today's offerings within the IT industry shows that SLAs for different service building blocks are put together without translating them into customer relevant value propositions. Imagine a hosting service including the delivery of data over a network. Instead of defining an SLA for availability at the customer's site, two figures, one for network availability, one for server availability are commonly given. Techniques like quality function deployment [12], a long standing engineering method to map product functionality to customer needs to develop products/services, do not enhance this situation, because they solidify the separation of customer needs into separate solution building blocks. Filling the matrix and adding up all the components does not lead to a complete end-to-end service description. It merely gives the requirements list for the engineers who build the product or compose the service.

To illustrate these shortcomings, consider the following examples taken from real world SLAs.

1. Giving details on “internal space management” and “Database administration” is commonplace in today’s SLAs. However, this information is not helpful to any prospective consumer of those service components. “We do all necessary maintenance and administrative work” would be a commitment that suffices to underline that the customer will not be bothered with technical issues of application management.
2. Consider the following original text from an SLA: “This service module includes provision, operation, rollout and updating of Lotus Domino databases and Lotus Domino mail databases.” The user might wonder what Lotus Notes can be used for. He/she should rather be told that “the email program will be Lotus Notes (which has a different look and feel than Microsoft products).” Constraints like restricted database size should be mentioned and explained in a user-friendly way i.e. “Storage space is restricted to 100 MB which is equivalent to about ten thousand emails without attachments.”
3. The SLA-phrase “Service transfer point is the server system concerned.” declares that the provider does not care how the service is delivered to the customer. Instead, delivery of the service is a different service that is described elsewhere within the description. This connection must be transparent for the customer. Better yet it should not be necessary because the transfer point is the customer’s screen.
4. “LAN availability 99.97%” is another very commonplace quality parameter. It is very important that the local network is available almost always, but how much is 0.03% downtime? Is that an hour per year or per week? And how does this affect the availability of the application I need to use? Can I use SAP and Word during that time?
5. “Response time for SAP data record read <4sec”. This SLA does not tell the user that he will experience an average of 4 seconds of response time by the system but that the server will need that long to look up a record in the database. How much time lies between a click and the visible response on the screen is not specified.

These examples show that firstly that SLA-authors rarely consider whom they are addressing with what kind of information and secondly all information is inherently focused on single resources. Of the five SLAs we have analyzed, not a single one attempted to bridge the gap between the resource-oriented, inside perspective and outcome-oriented, outside perspective. The impression that SLAs are resource focused is further supported by sources such as [13] which try to give business relevant advice on how to construct reliable SLAs. The customer as the target audience of an SLA is never considered in those expositions. Other literature that is interested in describing services is not focused on the service consumer either. The literature on web services for example explores ways of describing such services that are targeted at programmers or even machines [14].

The exposition in this section has focused on the weaknesses that service descriptions exhibit as means of communicating with the customer. We acknowledge that certainly there must also be a technical, resource oriented description of each service delivered as an inside definition of the service portfolio building block. However, an outside perspective must be defined and inside and outside perspective must never be mixed. A crucial factor to achieve user-oriented service descriptions is the strict separation between the manufacturing or input factor perspective and the value delivery or customer perspective. In this work we specify how one aspect of the separation between inside and outside view of manufacturing IT services can be done. We show how the description for the customer should be constructed. For the inside perspective see for example [15].

Arguing for customer oriented SLAs is quite easy if one takes it as given, that businesses want to make their offering transparent to the customers. While better communication is one possible way of differentiation in the market, there are also more subtle advantages of using descriptions of value propositions instead of descriptions of input factors to communicate with the customer. Firstly, customers have a higher willingness to pay for a service than for a commodity input due to the transfer of risk from the customer to the provider. Secondly, an integrated service offering decouples input and output and thus price and cost. The seller can therefore optimize both sides independently with possibly positive impact on the bottom line.

3. Service Dominant Logic for IT service providers

‘Service dominant logic’ as a new paradigm was first put forward by [9]. They promote a shift of focus from value creation through the exchange of goods to value co-creation through interaction between organizations and customers. Several key principles that mark the difference between goods and service focused thinking were identified by the authors. Below, we transfer these principles to the realm of IT-services. Some principles from the original paper focus rather on economics than on business issues and are therefore omitted.

The application of specialized skills and knowledge is the fundamental unit of exchange. Traditionally IT service providers have provided hard- and software that has been used within an organization. In a service dominant logic, the units of exchange are not hardware and software but skills and knowledge (related to information technology), that help customers to reach their goals. IT service providers have to focus on skills and knowledge and use hard- and software as means of delivering these services.

Goods are distribution mechanisms for service provisioning. Even though hard- and software are not at the center of attention anymore in a service dominant logic, they are still of some importance. While hard- and software are not the fundamental unit of value exchange, they are the distribution mechanism for services and are therefore essential for delivery from the service provider's perspective. From the customer's perspective, hard- and software are of little interest since users care only for the delivery of certain services.

The customer is always a co-producer. When shifting from the perspective of creating value through exchange of goods to a perspective where value is created by applying certain skills and knowledge provided through a service for the customer, the customer becomes a co-producer of value. With customers as co-producers, knowledge about the customer's processes becomes extremely important in order to provide services that can be applied within the customer's processes.

The enterprise can only make value propositions. With customers being co-producers, it is obvious that without the customers' interactions no value can be created. This also leads to the realization that organizations can only make value propositions. It depends on the customers to determine the value of the service and realize it in a process of co-creation.

A service centered view is customer oriented and relational. The shift from a goods dominant logic to a service dominant logic also affects the understanding of business relationships for IT service providers. Traditionally, goods-dominant logic is transaction-oriented while service-dominant logic requires customer- and relation orientation. This requires service providers to identify and define necessary processes, that facilitate this change in the business relationship (i.e. sophisticated controlling systems to implement pay-per-use models).

These principles form the basis for a mind-change within IT-service providers towards service orientation. Table 1 is based on [9] and shows more clearly the change in perspective when moving from a goods-dominant logic to a service-dominant logic.

The change in perspective and the shift to a new dominant logic are essential steps towards customer-oriented service descriptions. By focusing solely on services and how they are delivered to the customer, the focus shifts from single instances of hardware and software towards the total combination of hardware and software that is used to deliver a service. This also leads to a change in understanding of service delivery responsibility which we call the end-to-end perspective of the IT service delivery chain.

Table 1: Comparison of goods based versus service based IT providers.

	Traditional perspective of goods-centered IT service providers	**Emerging perspective of service-centered IT service providers**
Primary unit of exchange	Hardware Products and Software Licenses.	Knowledge and skills (possible embedded in hardware, software) are applied in the form of services.
Role of goods	Hardware and Software are developed, configured and installed.	Hardware and Software are used to deliver services but provide no value in themselves.
Role of customer / user	Customers are the receivers and users of hardware and software.	Customers are co-producers who create value by using the service in their processes.
Determination and meaning of value	Value is created through the exchange of hardware, software and project deliverables.	Value is co-created in a co-production process between IT-service provider and customer. Value can only be determined by customers and realized in their process.
Firm-customer interaction	Customers are acted upon. Hardware and Software is delivered and support is given to the customer.	The services are continuously provided to enable the customer's processes. Customers are operant; they participate in delivery of the service through its use.

4. End-to-End View on IT Services

End-to-end design is the design principle that has its roots in the design of distributed computer systems [16]. The principle suggests that the definition of requirements for a communication system requires an understanding of the end-points of this communication system. The requirements on the communication system for a tele-surgery application for example are different from those for a video-conferencing application. They require different control mechanisms at different levels of the communication or application system, respectively.

The end-to-end argument is an important part in the design of modern communication networks but it is not just limited to this context. The essential tasks when considering end-to-end design is the identification of the individual end-points. Applying the end-to-end principle to the IT service delivery chain, the end-points are the IT service provider on one side and the customer on the other side. With this perspective it becomes obvious that different components are involved when delivering IT services. All these components have to be included in the specification of service delivery parameters to ensure the delivery of value propositions according to the agreed upon service levels. Figure 1 depicts this end-to-end perspective of IT services which is not limited to the infrastructure at the IT service provider, but also has the different elements between IT service provider and end-user (i.e. WAN/LAN components, personal computer of the customer) in view.

The service levels for individual hard- or software components become irrelevant for customers, since from their perspective only the service level of the complete service is relevant. The essential change in perspective here is not only to recognize that there are different components involved in the service delivery chain, but rather that it is necessary to be in control of them to deliver a certain quality degree of value proposition to the customer. In that case, SLAs could still govern inter-provider relationships along the service value chain whilst the customer shall only be confronted with the end-to-end description.

These two concepts, the service-dominant logic and the end to end perspective on the IT-service delivery chain form the basis for the understanding of the changing role of IT-service providers and the need to adopt service descriptions accordingly. In the following sections we will outline a way of describing services that is more appropriate for communicating with the customer than nowadays service descriptions without forfeiting the detail and binding character of common SLAs.

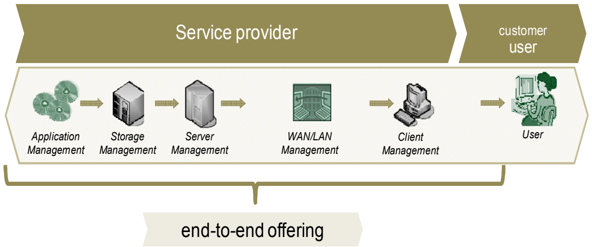


Figure 1: End-to-end offerings include the whole service creation process.

5. Research Process and Methodology

The research described in this paper follows "design research" guidelines as promoted (formerly as "design science") by [17] or [18]. In this section we are going to describe the overall research process as well as the design part of this research project which was carried out in cooperation with a large German IT-service provider (ITS henceforth).

One of the project's goals within the development of a prototype ERP System for a provider of IT-services was a suitable description of end-to-end services. As our analysis showed, current service descriptions with their SLAs are not able to provide this functionality due to their technical character, their use of complex terminology and their focus on input factors into IT-services rather than the business relevant effects.

The service to be offered within the prototype application was an end-to-end accounting service consisting of hosting, bandwidth, a desktop computer and all necessary supporting services as shown in figure 1. From our SLA analysis and multiple in-depth interviews with an IT Manager we had identified the need to create IT product offerings that focus on user needs and integrate several elements of the value chain. Assuming the ability to create the building blocks of such products as given (since they are being sold today), we identified our research question: Which rules should govern the design of user oriented descriptions of IT service bundles?

For theoretical grounding we reviewed literature on service level agreements [11], service orientation [19, 20] and service-dominant logic [9]. Additionally we analyzed the documentation of a major ERP system in order to complement our SLA analysis with a product manual that is explicitly focused on the user.

In order to understand the structural aspects of descriptions we built prototypes with a professional authoring tool for product descriptions which is being used to create manuals for complex machinery and analyzed the design specification of BMEcat [21], an industry standard for structuring XML Product descriptions. We also considered the available web-service specifications [14, 22, 23].

The next step involved the first cycle of an iterative description-engineering process. We built a model description by creating a document that described the IT-Service "End-to-End Accounting" which involved all necessary elements from application hosting to a desktop computer. After several iterations of design and prototyping, we informally evaluated and improved that description in a workshop with six IT professionals. After numerous improvements and internal review sessions we created a questionnaire for a field test, this time with IT-professionals who had not been exposed to our service description before. We presented the description together with a questionnaire to four accountants and one product manager of major European companies. The feedback was very positive and especially the product manager pointed out several aspects of our description that he found to be superior to the average service description he received as a basis for his buying decision.

After minor changes due to the interview results we presented our prototype and the rules to two IT product managers. Together with these managers we then created a new product description that was to be used within an ITS project. We started with the prototype and extended it to fit in the project context. After we had completed that task we had a real world product description that contained a considerable amount of details and was conforming to the design rules we had established with our prototype. The timelines of the research activities are shown in Table 2.

Table 2: Interviews and Workshops conducted to improve and test our artifact.

June 2007	First draft of product description, 2 day workshop with one IT professional from ITS
August 2007	Workshop with 6 professionals from IT organizations of 3 large firms
September 2007	Interview with two professional accountants from ITS
September 2007	Interview with two professional accountants from an ITS customer
October 2007	Interview with a procurement manager of a SME
January 2008	Workshop with two ITS product managers of IT services

6. The Design Rules

In the following we present the conceptualization of the findings within our research project. We do this by giving a set of design rules that should govern the creation of customer-oriented SLAs and service descriptions. The principles laid out in Table 1 can be considered as the transfer agents that help to make the transition from a set of ordinary SLAs to the type of customer oriented service description we propose in this paper.

Throughout the following text we will use an example of an end-to-end service for commercial managers. We will suppose that one provider is responsible for all elements of the service delivery chain. This approach eliminates some complications like the coordination of a chain of interdependent providers. One way to picture this "all in one" IT service provider is to think of it as the service provider of a large company. It has to buy all input factors and then delivers the integrated service to its customers. The rules, that should govern the design of a customer oriented description, are the following:

Rule 1. Outcomes define content. This most important rule for the creation of user oriented service description states that the leading principle for describing a service offering is outcome orientation. How the IT-provider produces the value propositions e.g. the potential to work on a workstation, is of no importance from the users point of view. Not the provisioning process of hard- and software is in focus, but the value that can be co-created. For example, the phrase "The software lets you create accounts and manage them" is acceptable while the phrase "on our blade servers we host SAP R/3 for you" is not.

Rule 2. Outcomes define the structure. Different from input-oriented SLA structure, service descriptions should be structured along dimensions of outcomes. The reader does not seek information on the building blocks needed to deliver a service but needs to know in which of his/her processes the service can be used. One should not describe accounting software and a database as separate entities but focus on accounting and controlling as tasks that are supported. Consequently, the table of contents of a written service description will not contain the basic elements like "ERP software", "hosting" and "Internet connection", but much rather "accounting", "controlling" and "annual accounts".

Besides these two very general rules there are more specific rules that provide a guideline for the creation of good service descriptions.

Rule 3. Quality parameters are defined end-to-end. End-to-end orientation of service descriptions requires the definition of customer oriented quality parameters. The parameter service setup time includes all tasks that have to be performed before the user can actually use the service. The setup time for a single component is irrelevant. With all quality parameters being focused on the user, effects of errors have to be observed from the user's point of view. Data loss probability and backup interval are therefore translated to 'hours of lost work'. Response times are measured at the client computer and not for some resource along the delivery chain. Technical parameters that have an influence on the response time line LAN latency need not be communicated to the customer.

Rule 4. Changes to the service potential are services themselves. Similar to goods that are delivered, services need to be set up. To start, change or stop a service, additional services, which carry out the appropriate functions, are necessary. These services need descriptions just as the core services. Stopping a service consists of picking up your hardware, returning all stored data and deleting your personal information. All these service elements create value for the user. Picking up the desktop frees space and returning data enables the customer to abide by the law that expects you to store accounting data for several years**.**

Rule 5. The description is binding and declared in a user / process oriented language. Services are to be described in a customer-oriented way that helps the customer to understand how the service creates value for him/her. However, the statements within a service description must also have the binding character of commitments. Short precise statements about the process support are desirable with the employed language coming from the targeted realm of application. While an accountant knows what 'SAP-GUI' is and a computer artist is familiar to a 'tile cache', both do not necessarily understand the meaning of "OSI layer 7" within a service description.

Rule 6. Cooperation is a defined process. As value is created in a cooperative process, customer and provider have to define how they work together. Instances of this rule are the regulation of service provider access to resources like PCs or printers that are located on the customer's premises. Furthermore, it must be made clear to the customer which actions the service can perform on its own and for which tasks it can only provide support. Closing all accounts is an automatic process, while ensuring their correctness cannot be guaranteed by the service beyond the correct use of math.

Rule 7. Information requirements are defined. All data needed for order processing should be requested during the ordering process. Therefore, a service description has to specify all data that needs to be known by the supplier in order to be able to deliver the service. However, this data must be within the knowledge domain of the purchaser: Data like the IP-address, application version numbers or system specifications are certainly not. This rule very simply avoids going back to the customer and asking the same question several times. Imagine situations where first the user management asks the customer for the exact location of his desk within the building and then the hardware delivery department calls to get exactly the same information. This would certainly reduce the perceived value of the service since the co-creation process of setting up the service involves more customer actions.

Rule 8. Dependencies are explicit. As a consequence of defining services in line with business processes, the possibility to order a service is dependent on the existence of other services, as explained above. Therefore the description of a service has to specify the dependencies on other services. In contrast to the order process of goods it is not only relevant which services can be ordered but also which services are already in use at the customer's site. As an easy example consider a cancellation service that can only be ordered once per corresponding service and cannot be ordered before the corresponding service has been ordered. This is different in an input oriented SLA. Nobody keeps you from ordering software without having a PC even though you cannot derive any value from it.

Rule 9. Structure goes from general to detail. As explained above the service offered is end to end. This means the scope includes all elements that are necessary to deliver a certain service to the customer. This includes a mouse as well as a LAN, hosting etc. In order to be able to describe such a service one could, as current day SLAs do, add up all elements that contribute to the service. Then the description would start by describing the mouse, go on to describe the LAN and end with the hosting service. For customers a different approach going from general to detail is better suited. On the highest level of abstraction, "IT-assistance for commercial managers" is offered. This service includes sub services like "IT-assistance for accounting", "IT-assistance for controlling" and "IT-assistance for office tasks". On each level of abstraction the value proposition is clear. On the topmost level the IT-service supports commercial managers. If more specific information is needed, the next hierarchy level tells the reader that accounting is part of that service. Within accounting the reader then gets some specifics about the accounting service. Its end-user oriented availability, the software deployed and the response times when using the user interface.

There are two notable aspects in the explanation above. Firstly, the description has a hierarchy that is as deep as desired by the writer. One can start at a very abstract level that puts the service description in a nutshell and get down to so much detail at the bottom of the description that one could describe every single transaction possible in an accounting system. Secondly, the information that is contained within the structure always pertains to the utility that the user can create by using the service. On the topmost level the value is 'getting work done'. At the most detailed level this utility could be 'generate printouts of transactions. On each level the description of the value creating service is accompanied by crucial information such as availability or quality parameters such as screen size or response time.

The hierarchical structure is a key element of making the description accessible to human readers. We tried different structures such as first describing all the functions before describing all the quality parameters or describing the service by adding up all elements in consecutive order. The accessibility of the information is severely reduced by such approaches.

Rule 10. Each service element is well defined. Describing all services in the same structure helps the reader to find information and ensures completeness of information. Therefore each service element should be described by the same categories of information:

- Overview: a short text describes as an abstract the main content of the service.
- Order Data: This includes the ordering number and name of the service, the minimum purchase quantity and the runtime of the service.
- Prices: They might consist of separate elements like setup charge and usage fee.
- Order information: As explained above all necessary data for processing an order has to be declared during the order process. Which data is needed is stated for each service element.
- Obligation to co-operate: An exact definition of the customer's tasks in value creation.
- Quality parameters: Are defined for every service element and always targeted at the end user.
- Dependencies: Specification which services have to be ordered together and which services have to be ordered, before another service.
- Additional information like target group of the offering, current technical solutions, information about resources and other trust building information.

Rule 11. Modularity is crucial. Having described each service in a user oriented manner with each service element containing all necessary declarations and being located in the right spot in the hierarchy, the description of the whole service portfolio is easily accessible to the reader. The structure can even be used to generate role-specific views of the description. Depending on the intended audience, specific elements of the descriptions can be omitted. User with low access rights for example can be excluded from seeing the price-elements of the descriptions while a manager would only see the overviews and price elements. For this to work it is necessary that all information is in the right place and marked correctly as specified under rule 10.

Figure 2 shows an example of an end-to-end product description. It is structured according to the presented rules and contains the appropriate information. The rules are implemented in the following way:

Rule 1 The outcome “functioning PC” is described rather than the process of performing tasks that are aimed at that outcome.

Rule 2 From the description it is clear, that not the delivered hardware is described but the functionalities that are available, such as a shared drive.

Rule 3 The quality parameter declares that within five days from ordering, one can use the computer and access all promise functionalities.

Rule 4 Not given in example.

Rule 5 The language is clear, concise and service levels such as “five days” are made explicit.

Rule 6 The customer’s tasks are clearly stated.

Rule 7 The data the customer has to provide is clearly stated.

Rule 8 The dependencies are separately stated. A frame contract must be established, before this product can be ordered.

Rule 9 Not shown. Picture the whole description as a tree. Below the element shown here there are more detailed descriptions available. The shown description is supplemented with additional details like technical information.

Rule 10 The structure of table 2 is repeated in every element of the description.

Rule 11 Table 2 only shows a very high level aggregation of the service provided. More detail is given further down in the description hierarchy (not shown here).

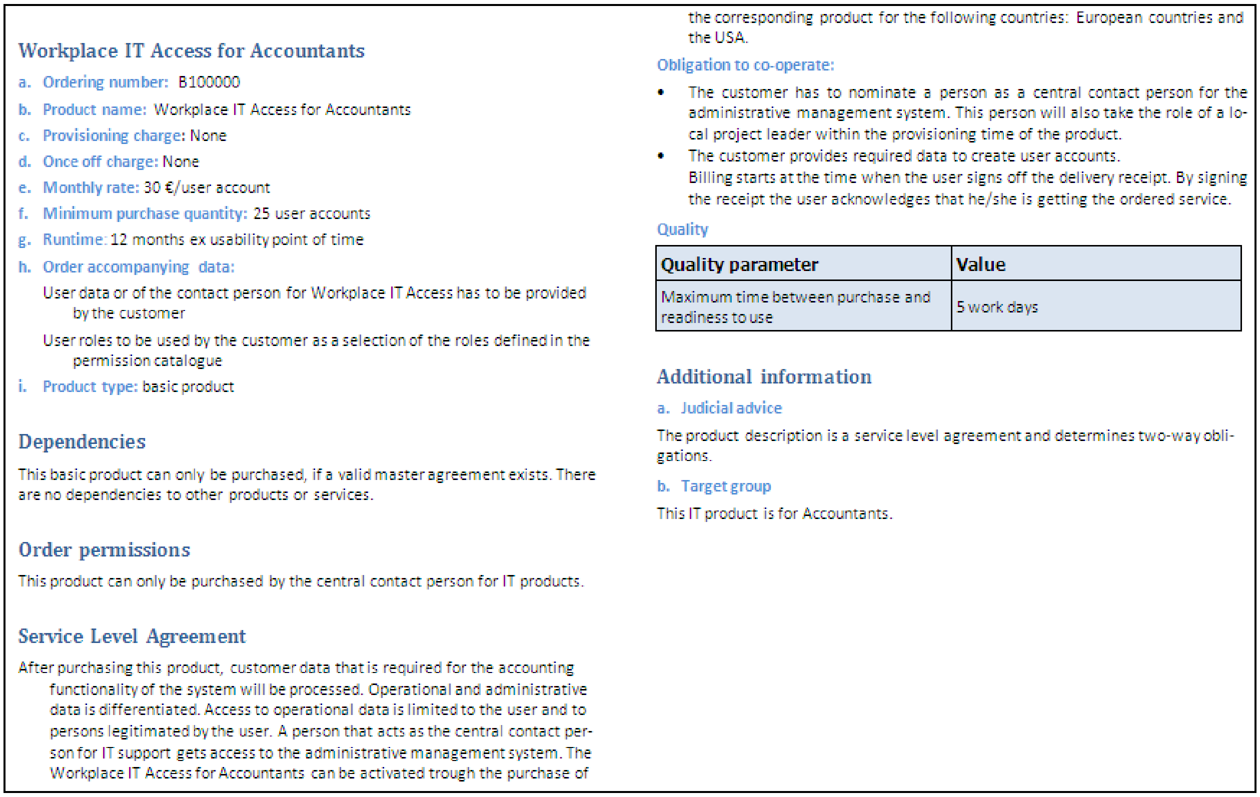
7. Conclusions

In the beginning of this paper we motivated our work by illustrating the lack of customer orientation in the IT industry. We showed this through the analysis of several SLAs that were resource focused instead of customer focused. Then we explained the sources we used for the construction of our prototype description. We drew from service science, from marketing, from the literature on service descriptions and SLAs as well as from works on industrialization. The description of the research process is quite extensive so as to show that theoretical as well as practical considerations have led to the artifact and the rules proposed above.

The results of our research process are design rules to describe customer focused IT services, that are tested through applying it in a description prototype. The most important achievement of this prototype is that due to the rules the whole document is focused on customer needs and not on technical inputs.

The advantage of such an approach lies in the communication to the customer. Firstly the customer of the service directly sees the value proposition of the service and does not have to compose different products and services into his/her personal value proposition. Thus we improve the value for the customer by lowering his transaction costs. At the same time, however, we do not forfeit the binding character of the service descriptions. All information necessary to judge the business relevant availability and quality of the service is declared. Furthermore this value-focused way to describe services offers an important advantage in a market for commodities. With only few possibilities to differentiate through features of input factors, the composition and communication of a service become more important. If a service provider succeeds in understanding the customer's needs and can create a spot on solution for him, the description will reflect this and thus offer an advantage over competitors.

Commoditization of IT services will force service providers to differentiate on other aspects than on price if they want to stay in the market. The proposed design rules to declare end-user oriented service descriptions concretize an unprecedented way to archive such an advantage.



Workplace IT Access for Accountants

a. Ordering number: B100000
b. Product name: Workplace IT Access for Accountants
c. Provisioning charge: None
d. Once off charge: None
e. Monthly rate: 30 €/user account
f. Minimum purchase quantity: 25 user accounts
g. Runtime: 12 months ex usability point of time
h. Order accompanying data:
User data or of the contact person for Workplace IT Access has to be provided by the customer
User roles to be used by the customer as a selection of the roles defined in the permission catalogue
i. Product type: basic product

Dependencies

This basic product can only be purchased, if a valid master agreement exists. There are no dependencies to other products or services.

Order permissions

This product can only be purchased by the central contact person for IT products.

Service Level Agreement

After purchasing this product, customer data that is required for the accounting functionality of the system will be processed. Operational and administrative data is differentiated. Access to operational data is limited to the user and to persons legitimated by the user. A person that acts as the central contact person for IT support gets access to the administrative management system. The Workplace IT Access for Accountants can be activated trough the purchase of the corresponding product for the following countries: European countries and the USA.

Obligation to co-operate:

- The customer has to nominate a person as a central contact person for the administrative management system. This person will also take the role of a local project leader within the provisioning time of the product.
- The customer provides required data to create user accounts.
Billing starts at the time when the user signs off the delivery receipt. By signing the receipt the user acknowledges that he/she is getting the ordered service.

Quality

Quality parameter	Value
Maximum time between purchase and readiness to use	5 work days

Additional information

a. Judicial advice

The product description is a service level agreement and determines two-way obligations.

b. Target group

This IT product is for Accountants.

Figure 2: Excerpt of the prototype description.

References

[1] Carr, N., *IT doesn't matter.* IEEE Engineering Management Review Online, 2004. 32(1): p. 24-24.

[2] Carr, N.G., *The End of Corporate Computing.* Mit Sloan Management Review, 2005. 46(3): p. 67-73.

[3] Böhmann, T., M. Junginger, and H. Krcmar. *Modular Service Architectures: A Concept and Method for Engineering IT Services*. in *Proceedings of the 36th Hawaii international Conference on System Sciences (HICCS'03)*. 2003.

[4] Zarnekow, R., W. Brenner, and U. Pilgram, *Integrated Information Management. Applying Successful Industrial Concepts in IT*. 1 ed. 2006, Berlin: Springer.

[5] Zarnekow, R., *Produktionsmanagement von IT-Dienstleistungen. Grundlagen, Aufgaben und Prozesse*. 2007, Berlin: Springer Verlag.

[6] Peppard, J., *Managing IT as a Portfolio of Services.* European Management Journal, 2003. 21(4).

[7] Trienekens, J.M., J.J. Bouman, and M. van der Zwan, *Specification of Service Level Agreements: Problems, Principles and Practices.* Software Quality Journal, 2004. 12(1).

[8] Vargo, S.L. and R.F. Lusch, *Service-dominant logic: continuing the evolution.* Journal of the Academy of Marketing Science, 2008. 36(1): p. 1-10.

[9] Vargo, S. and R. Lusch, *Evolving to a New Dominant Logic for Marketing.* Journal of Marketing, 2004. 68(1): p. 1-17.

[10] Sturm, R., W. Morris, and M. Jander, *Foundations of Service Level Management*. 2000: Sams.

[11] Pietsch, W., *Kundenorientierte Ausgestaltung von IT Service Level Agreements.* Software Process Improvement: 12th European Conference, EuroSPI 2005, Budapest, Hungary, November 9-11, 2005: Proceedings, 2005.

[12] Sullivan, L., *Quality Function Deployment.* 1986.

[13] Soebbing, T., *Handbuch IT-Outsourcing.* Rechtliche, strategische und steuerliche Fragen. Redline Wirtschaft bei Ueberreuter, Frankfurt Wien, 2002. 19.

[14] W3C, *Web Services Glossary.* www.w3.org/TR/ws-gloss/.

[15] Ebert, N.U., Falk; Hochstein, Axel; Brenner, Walter, *A Service Model for the Development of Management Systems for IT-enabled Services.* Proceedings of the Thirteenth Americas Conference on Information Systems (AMCIS 2007), 2007.

[16] Saltzer, J., D. Reed, and D. Clark, *End-To-End Arguments in System Design.* Technology, 1984. 100: p. 0661.

[17] Hevner, A., et al., *Design Science in Information Systems Research.* MIS Quarterly, 2004. 28(1): p. 75-105.

[18] Peffers, K., et al. *The Design Science Research Process: A Model for Producing and Presenting Information Systems Research*. in *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006)*. 2006.

[19] Maglio, P.P., et al., *Service systems, service scientists, SSME, and innovation.* Communications of the ACM, 2006. 49(7): p. 81-85.

[20] Papazoglou, M.P., *Service-oriented computing: concepts, characteristics and directions.* Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on, 2003: p. 3-12.

[21] Schmitz, V., O. Kelkar, and T. Pastoors, *Specification BMEcat, Version 1.2.* URL: http://www. bmecat. org, 2001.

[22] Alonso, G., *Web Services: Concepts, Architectures and Applications*. 2004: Springer.

[23] Casati, F., et al., *Business-oriented management of Web services.* Commun. ACM, 2003. 46(10): p. 55-60.