

Specification of service level agreements, problems, principles and practices

Citation for published version (APA):

Trienekens, J. J. M., Bouman, J. J., & Zwan, van der, M. (2004). Specification of service level agreements, problems, principles and practices. *Software Quality Journal*, 12, 43-57.
<https://doi.org/10.1023/B%3ASQJO.0000013358.61395.96>,
<https://doi.org/10.1023/B:SQJO.0000013358.61395.96>

DOI:

[10.1023/B%3ASQJO.0000013358.61395.96](https://doi.org/10.1023/B%3ASQJO.0000013358.61395.96)
[10.1023/B:SQJO.0000013358.61395.96](https://doi.org/10.1023/B:SQJO.0000013358.61395.96)

Document status and date:

Published: 01/01/2004

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.



Specification of Service Level Agreements: Problems, Principles and Practices

JOS J.M. TRIENEKENS

University of Technology Eindhoven and KEMA Quality B.V., The Netherlands

j.j.m.trienekens@tm.tue.nl

JACQUES J. BOUMAN

University of Technology, Eindhoven, The Netherlands

MARK VAN DER ZWAN

Improve Quality Services, The Netherlands

Abstract. Software intensive systems (SIS) increasingly influence the ability of enterprises to be competitive in continuously changing market situations. The integration of these systems into organizations, and in particular the subsequent exploitation, maintenance and service activities, have become of utmost importance. Unfortunately the area of exploitation and operation, also known as service management, is still rather immature. Service management covers services such as performance and availability support, end-user and help desk support, education, and maintenance. One of the main concepts of service management is the Service Level Agreement (SLA). The goal of an SLA is to bridge the gap between service provider and users or customers. However, there exist many problems and unsolved questions regarding the specification and the quantification of SLAs. This paper addresses the specification of SLAs on the basis of three well-founded service management principles, respectively: ‘continuity in service management,’ the pit/shell principle of a service, and the principle of specifying the quality of both a service process and a service object. Finally, the paper addresses the validation of these principles in practice.

Keywords: service level agreements, service object, service process, quality characteristics, case study results

1. Introduction

Software intensive systems (SIS) increasingly influence the ability of enterprises to be competitive in continuously changing market situations. Over the last ten years the primary processes of many organizations have become strongly dependent on them. As a consequence the integration of these systems into organizations, and in particular the subsequent exploitation, maintenance and service activities, have become of utmost importance. Unfortunately the area of exploitation and operation, also known as service management, is still rather immature in comparison with other SIS areas, such as for instance systems analysis and design (Robertson and Robertson, 1999). Although many structured methods and techniques exist for the structured development of SIS there are hardly any approaches and instruments for the specification of services for SIS.

Regarding integration and exploitation the relatively new domain of service management has emerged rapidly. This is amongst others reflected by the amount of effort and resources that are spend on service management. Service management covers services such as performance and availability support, end-user and help desk support,

education, and maintenance. Organizations often need various kinds of specialised support during the operation and the exploitation of their SIS. It is estimated that large companies spend up to 50% of their investment portfolio on SIS related investments (Renkema, 1997). Generally it is assumed that only 20% from these investments are directly spent on SIS development. The remaining 80% consist of the costs of integration, maintenance and exploitation of SIS. These costs are referred to as service costs.

Over a large number of years important work has been carried out by CTTA (1987). A collection of best practices, which is called Information Technology Infrastructure Library (ITIL), focuses in particular on the investigation of the various aspects of Service Management and introduces as one of the main concepts for Service Management the Service Level Agreement (SLA). However, the quality of services in service level agreements and its metrification are not addressed. In the Capability Model for IT-enabled Outsourcing Service Providers (eSCM) (Siegel and Gopal, 2002) a framework and an improvement path is described that customers should expect service providers to progress along from the minimum level of having the capability to deliver a service that meets client requirements, up to the highest level of enhancing value through continuous innovation. In this Capability Model, which has five levels of improvement, customer requirements management and measuring and control of service processes are explicitly addressed. For example, on the maturity levels 2 and 3 it is stated that procedures for SLA specification, measurement and control should be developed and implemented.

Over the last years more and more publications on SLAs can be found in various journals and magazines, see, e.g. (McBride, 1998; Passmore, 1996; Sasser et al., 1978). One of the most important trends is the shift of the goal of an SLA from being a financial and technical contract towards an instrument for the management of the customer's expectations. However, there still exist many unsolved questions in the SIS Service Management area, e.g., regarding the definition of services and the specification of SLAs.

In approaches for assessing process capability (e.g., ISO15504 based on the SPICE project (Solingen et al., 2000)) the quality of (service) processes is the central issue. In this type of approaches process capabilities such as customer satisfaction, formal service contracts, etc., are assessed and are rated in a well-founded and structured way. Although well-defined terms can be found for (service) process quality concepts there are no methods or techniques given that support the specification and metrification process of service quality in service level agreements. Approaches that are more directly related to the specification of service quality are on the one hand well-known methods, from the general industrial manufacturing domain, for the identification and specification of customer requirements, e.g., Quality Function Deployment (QFD) (Cross, 2000), and on the other hand software analysis and design methods from the software requirements engineering domain, e.g. (Robertson and Robertson, 1999). In approaches of these types the requirements elicitation process, the collaboration process between engineers and customers, and the translation process of quality-from-a-customer perspective into quality-from-an-engineering perspective are extensively addressed. This paper is in particular inspired by these types of approaches.

This paper will focus on the basic concepts for defining and specifying SLAs. In Section 2 some main problems with current SLA specification approaches will be addressed. In Section 3 the basic concepts for SLA specification will be introduced. Section 4 reports about the application of these basic SLA concepts in a practical case study. Finally in Section 5 the main conclusions are given and future research topics are addressed.

2. Specification of SLAs: problems and suggestions for solutions

Service Management is a relatively new domain in SIS management (Ruijs et al., 2000). SIS Service Management focuses on the planning, monitoring and control of SIS services. Currently the SIS Service domain shows a rather scattered picture of various types of defined and undefined services. As a consequence it is difficult to manage SIS services in an effective and efficient way. This counts for all SIS services, such as recoverability services, performance planning services, help desk services, maintenance, educational services, etc.

Although there is much evidence about the role and importance of SLAs in the context of Service Management, SLAs show many shortcomings in practice. From empirical research major problems have been identified (Bouman et al., 1999; Trienekens and Kusters, 1992). Hereafter we will address them briefly.

Specification of effort versus specification of results

Most SLAs focus on agreements regarding the effort that will be spent by an SIS provider in case a problem occurs. However, no commitments are specified regarding the effectiveness of a service for a customer's business processes and his business objectives. Instead of "services regarding system X will support your company with reaching business objectives Y," an SLA usually states "we will be at the problem location within a certain amount of time in case system X breaks down."

Unclear service specifications

Agreements on for instance "the availability of a network" are usually specified using a metric called the availability percentage. It is often hard to determine what the precise meaning of such a metric is in the context of a specific business location. For instance, what is the difference between an availability percentage of 98% and 99%? And does 98%, on a yearly basis, mean that the network is allowed to go "down" for a whole week, after being "up" for the last 51 weeks?

Incomplete service specifications

It is difficult to make complete agreements on particular services, for instance on services such as security control and disaster prevention. A key problem is that it is often

difficult to describe and to quantify the consequences of fraude and disasters and to determine what type of services are needed and to what extent.

Insufficient cost management

Cost management is often expressed as “making agreements on a fixed price each year for an integrated set of SIS services.” In what way these costs can be differentiated and can be related to specific SIS services, in conformance with the needs of a customer, is often unclear. As a consequence it is very difficult to determine a price/performance optimum for each particular service for a customer.

“Dead-end” SLA documents

A Service Level Agreement is often a technical document regarding concepts and terminology that can only be understood by a small group of technology oriented specialists. Often, evaluation and improvement does not take place on a regular basis. Such a ‘static’ or ‘dead-end’ document has a very restricted meaning for end-users and their management. The latter are not able to interpret the ‘agreed’ service specifications nor are they able to tune or enrich the agreed service level agreements. After some time the specification of an SLA tends to become ‘just an unsatisfying tradition.’

In the foregoing list a number of major problems in SLA specification have been addressed. Hereafter we will point to suggestions that have been given in literature to solve these problems.

In (Steinke, 1997) a cost driven oriented approach is followed, stating that important objectives of an SLA are respectively better management of financial resources and increased confidence in the budgeting process. Other references address particular technical aspects of services, such as the definition of various service metrics (McBride, 1998; Passmore, 1996). The process of SLA development is in the given references only described in global terms. The following activities in SLA development are mentioned:

- The investigation of the business context of a customer and the identification of his needs.
- The determination of service levels by a provider.
- The obligation of a provider to come to a service agreement with a customer.
- The development of basic rules for the collaboration between a customer and a service provider.

In software management science the importance of a consensus based approach between stakeholders (e.g., customers) and developers is emphasized (Boehm and Ross, 1989), in particular regarding requirements determination. These concepts, such as ‘stakeholder win-win,’ are applicable to SLA specification, in particular to improve the usage of an SLA to serve as a communication aid and a conflict prevention instrument.

Having addressed some of the main problems in practice and some directions for solutions in literature we can conclude that there are not yet clear concepts and use-

ful techniques for the specification of service level agreements. To contribute to the solution of these problems we introduce in the following section first some basic principles for the specification of service level agreements. In Section 4 we will apply and validate these principles in practice.

3. Principles for specifying Service Level Agreements

This section presents three principles for the specification of SLAs. These principles are being considered as basic concepts. The principles that will be addressed hereafter represent lessons learned or understandings that have evolved from practical case studies (Trienekens and Kusters, 1992). The three principles are, respectively:

- The principle of continuity in SLA specification.
- The SLA pit-shell principle.
- The principle of quality of both SIS service processes and SIS service objects.

3.1. Continuity in SLA specification

General service management literature proposes that quality should be defined as “meeting and/or exceeding customers’ expectations” (Passmore, 1996; Rodriguez-Dapena et al., 2001). However, in many cases there exists a gap between what a customer needs and what a service provider is able or willing to provide (Parasuraman et al., 1986). This gap cannot be bridged in just one linear step. The identification of the needs of a customer, the design and implementation of a service process, and the evaluation and improvement of the service should be considered as an iterative and continuous process. For this continuous process a model has been developed that is called the Service Management Lemniscate, see Figure 1.

This model has to be interpreted as follows. The service needs of a customer have to be identified carefully, preferably in close collaboration between a customer and a provider (we come back to that in Section 4.1). Service needs have to be derived from the characteristics of the business environment in that the SIS performs, e.g.,

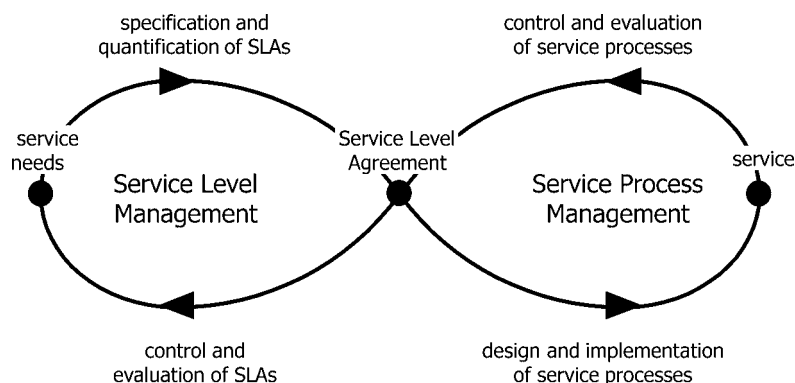


Figure 1. The Service Management Lemniscate.

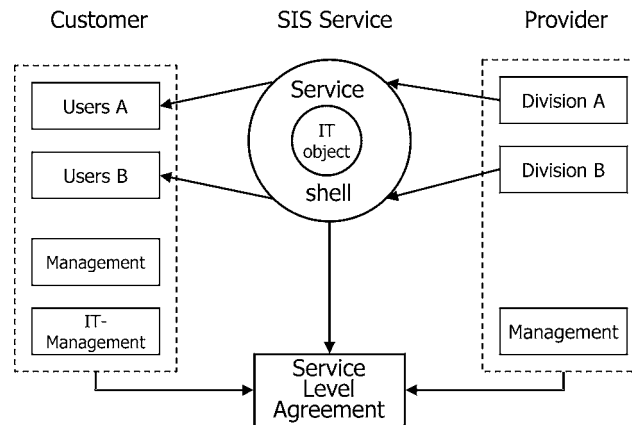


Figure 2. Context of an SLA.

with respect to the complexity and the dynamics of the business processes (Trienekens and van der Zwan, 1997). Subsequently the needs have to be translated into service requirements and specified in terms of Service Level Agreements. The SLA is the central point in the Service Management Lemniscate. An SLA supports the communication about services and forms a basis for the development and implementation of the service processes. Further the Service Management Lemniscate stresses the importance of evaluation and improvement, both of the service processes and the SLA itself. These concepts of evaluation, improvement and learning are receiving increased attention in the software industry (Boehm and Ross, 1989; Software Quality Institute).

3.2. SLA context and content (the pit and shell principle)

Figure 2 presents an SLA in its context, i.e., with respect to the parties that are involved in service level specification and the content of an SLA. This figure stresses the importance of recognizing the various stakeholders in an early phase of the SLA development process (Cross, 2000).

In Section 4 we will come back to this in the case study. Hereafter we address first the content of an SLA, respectively the SIS service object, the service process and the interrelations. Subsequently we will address the three different types of service objects that can be recognized.

Service pit and service shell

In service management there has often been confusion regarding an adequate definition of a SIS service. Both customer and provider are interested in both process and deliverable aspects of a service. To avoid confusion we suggest that both an object and a process aspect of a service should be recognized.

The service object is of primary interest for the customer, such an object should for instance be reliable and fast. The specification of a service will have to refer to

this service object but also to its surrounding business context (e.g., the stakeholders). We call this service object the kernel or the ‘pit’ of a service. Examples of a pit are respectively a software application, a hardware component of an information system, an infrastructural network component, etc.

The service process is of primary interest to the provider, e.g., the service process should be efficient and cost-effective. Examples of service processes are recoverability services, help desk services, education and training services, etc.

In practice one particular service can cover a diversity of SIS objects. For example, a SIS service such as ‘office automation’ covers a variety of SIS objects such as a desktop computer, a notebook, software applications, printers, etc. Of course one SIS object can also be subject to a variety of SIS services. In both situations the principle of ‘pit and shell’ is applicable, respectively a ‘multiple pit’ service situation and a ‘multi-layered shell’ situation.

Types of SIS service objects

In order to specify a SIS service precisely three types of SIS objects are distinguished, respectively the component, the system and the infrastructure. They are defined as follows:

- *SIS component*: An entity that supports the processing, the storage, the distribution, and the input and output of data.
- *SIS*: A coherent set of components that support the execution and performance of tasks of end-users in a particular business process.
- *SIS infrastructure*: A coherent set of components that form the (technical) basis for the operation and performance of multiple SIS.

In Figure 3 an example of the interrelations between the three types of SIS objects is shown. In this example three SIS can be distinguished that both consist of SIS components and SIS infrastructural components. The SIS infrastructural components are, per definition, always a component in more than one system (SIS).

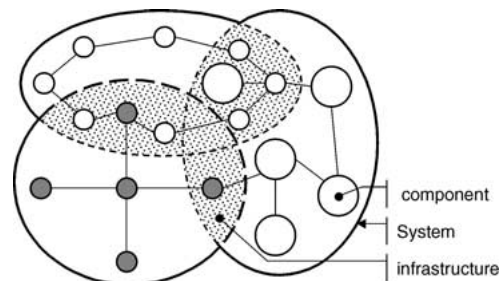


Figure 3. Three types of SIS service objects and their interrelations.

Table 1. Quality characteristics of SIS service objects and service processes

SIS service object & service processes	Quality Characteristics			
	Availability	Performance	User friendliness	Flexibility
Service object	Availability Rate			
Education			Acceptance Rate of change proposals	
User Support		Maximum Response time		
Change Management				
Maintenance				
.....				

3.3. SIS service process and service object quality

To determine the quality of a SIS service a distinction is made between the quality of a service process and the quality of a service object. Service process quality is in particular the responsibility of a provider and is 'effort related.' Service object quality is in particular the responsibility of a customer and is 'business result related.'

Table 1 presents the quality characteristics on the horizontal axes, respectively availability, performance, user friendliness and changeability. On the vertical axes the SIS object and the distinguished SIS service processes are given. The service processes are based on standardized descriptions derived from (CCTA, 1987). The quality characteristics are based on ISO9126 (ISO/IEC, 2001). This ISO9126 standard makes a distinction between quality in use (business environment oriented), external (customer oriented) and internal quality characteristics (engineering oriented). In this paper we refer to the external quality characteristics that act as a bridge between the other two types of characteristics and that give support to translate quality-in-use (i.e., quality as required by the business) into internal quality (i.e., quality to build in by the engineer). ISO9126 gives operational definitions of the three types of quality characteristics and gives a number of metrics to improve the applicability.

In Table 1 each cell contains metrics that can be used to quantify the agreements on a specified service (both of type object and process). Table 1 has as examples of service metrics, respectively: the *availability rate* for the quality characteristic *availability* of the *service object* 'notebook,' the *response time* for the quality characteristic *perform-*

ance of the *service process* ‘user support,’ and the *acceptation rate* for the quality characteristic *flexibility* of the *service process* ‘change management.’

Particular quality characteristics of both service objects and service processes have to be derived from the particular needs of users of SIS objects in a business system (Ruijs et al., 2000). For example, for a *service object* such as an ERP system that has to perform specific planning functions within a certain amount of time, quality characteristics such as *performance* and/or *availability* (during a specific period in time) have to be specified. But also regarding the needed *service processes* such as ‘user support’ the *availability* of this process and its *performance* could be addressed.

A set of agreements on the quality of both SIS service objects and service processes forms a basis for a Service Level Agreement. In the following section we will present a validation of the described service management principles in a practical case study.

4. Validation of SLA principles in practice, a case study at Eindhoven University of Technology

As part of its education renewal program the Eindhoven University of Technology (TUE) offers every first year student a notebook computer. The first notebooks were delivered to the 1997/1998 generation. At this moment every student at the TUE has a notebook, which will be used in lectures, during tutorials, in study groups, at home, etc. This means that currently more than 5000 notebooks are in use.

The ICT Service centre of the TUE (ICTS) was not suited for providing services to many end-users. Therefore a dedicated, new service centre has been set up to manage the services regarding the large number of notebooks. Typical tasks of this Notebook Service Centre (NSC) are the solving of hardware and software problems, the installation of new software and the support of students to use their computer more effectively and efficiently. The NSC should ultimately become a flexible organisation with a close and direct contact with its customers, capable of dealing with the expected large variety of demands, and able to anticipate new or changing service needs. Therefore a project has been developed to identify the service needs of the students, their lecturers, staff employees and other stakeholders, to specify service level agreements, and the stepwise evaluation and improvement of the services. In this project the stakeholders have been identified (being an important context aspect) and based on that the parties were selected that should play a role in the SLA specification process. We will address this process, including the application and validation of the SLA continuity principle, in Section 4.2. In Section 4.1 we will focus on the application and validation of the SLA pit/shell principle and the principle of quality of both SIS service objects and service processes.

4.1. Application of the pit/shell principle and the service quality principle

First the SIS objects have been investigated and defined. These objects include software applications, notebook (hardware) components, and external devices such as printers, scanners, cd-rom copiers, etc. Based on the characteristics of these ob-

jects, the characteristics of their usage and the service needs of the users the service processes have been determined and defined.

Second, the quality characteristics of both the SIS service objects and the service processes have been identified and specified. Regarding the availability of the service both the availability of the notebooks and the software applications (the service object or the ‘service pit’) and the availability of the service processes (the ‘service shell’) were marked as a high priority service.

Further, the importance of fast helpdesk support (response time/performance) was stressed and the flexibility regarding both the software applications and the service processes. Flexibility of the service process was in particular important because this process was new to all involved parties and it was impossible to predict precisely the usage of the notebooks and the requirements regarding the service process.

Consequently, the quality characteristics of the SLA have been made measurable. We give two examples, respectively the performance of the user support process and the availability of the service object notebook. Furthermore, we give a brief clarification of the structure of the SLA specification.

The quality characteristic performance

The performance of the user support process has been made operational by defining incident priorities and problem solution times, see Table 2.

The priority ranking is based on a distinction between the different types and different periods of usage (e.g., lectures, tutorials, self-study and examinations), and the severity and consequences of potential failures. For example, a student with a crashed notebook will get immediate support during an examination session. During a normal lecture period, a problem will be solved between 2 hours and 8 hours (at a maximum). Furthermore, the agreement has been made that:

- 95% of all reported incidents will be solved according to the minimal problem solving time;
- 99.9% of all reported incidents will be solved according to the maximum problem solving time.

The percentages are calculated per calendar month. Incidents that are not solved within the maximum problem solving time will be reported individually in a monthly service level agreement report.

Table 2. Metrification of the performance of the service process ‘user support’

Incident priority	Minimal problem solving time	Maximum problem solving time
1	1 hour	4 hour
2	2 hour	8 hour
3	4 hour	16 hour

Table 3. Metrification of the availability of the service object 'notebook'

Availability: the degree to which a notebook is available, in accordance with the specifications, during opening hours of the Notebook Service Centre.

Availability rate for an individual notebook:

$$A_n = \frac{(t_s - \sum_{i=1}^{I_n} D_i \cdot W_i)}{t_s} \cdot 100\%, \quad t_s > 0.$$

A_n : availability rate of notebook n .

t_s : service time.

I_n : number of Incidents per month of notebook n .

D_i : downtime of Incident i .

W_i : weight of Incident i :

$W_i = 1$ for Incident type 1;

$W_i = 0.6$ for Incident type 2;

$W_i = 0.1$ for Incident type 3.

The quality characteristic availability

The availability of the notebook (and its software applications) has been defined and consequently the metric availability rate has been specified, see Table 3.

In calculating the availability rate the severity of incidents is taken into account: the more severe an incident is (i.e., a higher weight), the more it negatively influences the availability rate. The service time is defined as an agreed period of time, during which the availability rate is calculated. This is the number of working days per calendar month times the opening hours of the Notebook Service Centre.

Regarding the specification and the metrification of the service objects and processes it should be stated that this is not a trivial exercise. Collaboration with the customer and consensus building, for example regarding the type of metrics that could be useful in the particular situation, are of utmost importance. This process of collaboration in defining metrics can be structured by industry models such as Goal Question Metric (GQM) (Trienekens and Kusters, 1992). GQM approaches, such as (Solingen et al., 2000), support practitioners in making their goals explicit regarding measurement, and to derive and define metrics that are needed to be able to decide whether goals are reached or not. GQM stresses the importance of a well-organized process for collaboration and consensus building between the different involved parties in defining measurement.

The structure of an SLA specification

The service processes, the SIS service objects and their metrifications have been specified in readable user-oriented SLA sections. Table 4 gives an example of the structure of an SLA. The clear-cut end-user oriented sections can simply be enriched or improved, without having to change the whole document.

4.2. The development process of the SLA

The SLA process has been divided into a number of phases, respectively Introduction (SLA principles and objectives), Needs, SLA specification and Implementation. In the

Table 4. Structure of the SLA specification

⋮
⋮
Section <i>x</i> . Service objects
3.1 Context (a.o. type of stakeholders, requirements)
3.2 Notebook (a.o. software, hardware, additional devices)
3.3 Quality characteristics
3.3.1 Availability
3.3.2 Performance
3.3.3 User friendliness
3.3.4 Flexibility
3.4 Metrification
Section <i>x</i> + 1. Service processes
4.1 Context (a.o. types of stakeholders, requirements)
4.2 The user support process
4.2.1 Parties and responsibilities
4.2.2 Metrification
4.2.2 Procedures
4.3 The change management process
4.3.1 Parties and responsibilities
4.3.2 Metrification
4.3.3 Procedures
⋮
⋮

Table 5. Effort in hours, spent by the parties during the different phases of the SLA development process

Parties involved	SLA development phases			
	Introduction	IT service needs	SLA specification	Implementation
Service Provider	7	10	28	24
SLA Expert	21	150	265	32
Contract Expert	2			188
User Representatives	2	2	2	
Total Effort (hours)	32	162	305	244
Timeline	September	October–December	January–March	April–June

first phase introductions have been given to inform the involved parties about the SLA principles and the objectives of the SLA development process. In this phase four parties have been recognized that would play a role during the SLA development process, respectively: the SLA expert, the TUE service provider, the contract expert, and the user representatives. Regarding the latter, the user representatives, both educational experts, the faculty director of education and representatives of the students were involved. In the second phase, Service Needs, it was initially assumed that the different faculties, such as computer science, architecture and chemical engineering would need different service agreements. Although some differences were found in their service needs the similarities were much more obvious. In phase three, the SLA specification phase, the service objects were classified and defined, the quality characteristics were specified and subsequently metrified.

In Table 5 the effort in hours is given that has been spent by the different parties during the SLA development process.

The table shows that it took quite some time for the SLA expert to identify the service needs and to specify the services, their quality characteristics and the metrics. In the last phase, Implementation, in particular, the finalization of the SLA contract took a lot of time. Legal experts needed more time than expected to fill in and justify all the details of the service level agreement.

4.3. Evaluating and improving the SLA: closing the service management lemniscate

The service level agreements have to be evaluated on a continuous basis. The results of these evaluations can serve as a basis for further improvements. Examples of improvements are the tuning and further elaboration of metrics, and the extending of the feed-back from customer to service provider (based on interviews with students). One of the consequences was that, instead of only one central service location, it was decided that each faculty should develop its own local helpdesk. Each faculty has then the freedom to tune its services to the local needs of their students (within an overall service framework) and to come up with suggestions for service improvement and service metrification.

5. Conclusions

Service level agreements for SIS, which are useful for both customers and providers, are rarely seen in practice. Reaching consensus between customers and providers on services to be delivered is a difficult and time-consuming task. This paper presents first steps towards a structured approach for the specification of service level agreements in order to speed up the process of service level agreement specification, and to improve the effectiveness of SLAs for both customers and users. Therefore three principles for service level agreements specification have been developed. These principles have been made operational and have been applied and validated in a practical case study. The first principle of continuity (the lemniscate) stresses the need for a step-by-step development and improvement of the SLA. It also offers a framework for the communication and the collaboration between customer and service provider. The lemniscate is understandable for each party and shows in one figure the bridging of the service gap between customer and provider. The second principle of the distinction between a SIS service object and a service process, as being two complementary service aspects, appears to be fruitful regarding the precise definition and specification of a service. The third principle of recognizing quality characteristics for both the SIS service objects and the service process shows that service level agreement should and can be specified in an objective and quantifiable way. The quality characteristics of a SIS service can be made measurable in a joint cooperation between customer and service provider in practice.

Although interesting results have been gained in our SLA research we state that there is currently a lack of formalism in the specification and metrification approach. For example, the definitions and the metrics on service quality characteristics have to be improved further. In our near future research on SLA specification and metrification we will strive in particular at the quantification of the presented service principles and practices.

References

- Boehm, B.W. and Ross, R. 1989. Theory-W software project management principles and examples, *IEEE Transactions on Software Engineering* 15(7): 902–916.
- Bouman, J.J., Trienekens, J.J.M., and van der Zwan, M. 1999. Specification of service level agreements, clarifying concepts on the basis of practical research, *Proceedings of the 9th International Workshop Software Technology and Engineering Practice*, S. Tilley and J. Verner (eds.), IEEE Computing Society, Los Alamitos, pp. 103–111.
- Cross, G.A. 2000. Collective Form, an exploration of large-group writing, *The Journal of Business Communication* 37(1): 77–100.
- CCTA. 1987. *Information Technology Infrastructure Library*. London, HMSO Publications Centre.
- Herzwurm, G., Mellis, W., and Schockert, S. 2000. *Joint Requirements Engineering Software Development*. San Mateo, CA, Morgan Kaufman.
- ISO/IEC 9126-1. 2001. Software Engineering—Product Quality—Part 1: Quality Model, <http://www.iso.ch/iso/>
- McBride, D. 1998. Successful deployment of IT Service Management in the distributed enterprise, Hewlett-Packard Company, White paper, doug_mcbride@hp.com
- Parasuraman, A., Zeithaml, V., and Berry, L.L. 1986. SERVQUAL: A multiple item scale for measuring customer perceptions of Service Quality, Marketing Science Institute, Cambridge, MA.
- Passmore, D. 1996. Setting performance expectations, *Business Communications Review* 26(12): 20–22.
- Renkema, Th.J. 1997. *Investments in Information Technology Infrastructures*. Kluwer Bedrijfsinformatie.
- Robertson, S. and Robertson, J. 1999. *Mastering the Requirements Process*. Addison-Wesley.
- Rodriguez-Dapena, P., Vardanega, T., Trienekens, J., Brombacher, A., and Gutierrez, J. 2001. *Nonfunctional Requirements as a Driving Force of Software Development*. Software Quality Professional.
- Ruijs, L., Niessink, F., and Trienekens, J.J.M. 2000. *Towards Mature Service Management* (in Dutch). Academic Service.
- Sasser, W., Earl, R., Olsen, P., and Wyckoff, D.D. 1978. *Management of Services*. Boston, MA, Allyn & Bacon.
- Siegel, J. and Gopal, A. 2002. eServicesCapability Model (eSCM) v1.1, Carnegie Mellon University, <http://itsqc.srv.cs.cmu.edu/escm/>
- Software Quality Institute at Griffith University, Australia, <http://www.sqi.gu.edu.au/SPICE/>
- Solingen van, R. and Berghout, E. 1999. *The Goal/Question/Metric Method, A Practical Guide for Quality Improvement of Software Development*. New York, McGraw-Hill.
- Solingen van, R., Berghout, E., Kusters, R.J., and Trienekens, J.J.M. 2000. Learning in software process improvement, *Journal for Information Systems Technology*.
- Steinke, S. 1997. Taking charge of service levels, *Network* 12: 77–81.
- Trienekens, J.J.M. and Kusters, R.J. 1992. Customer orientation as a basis for computer supported technologies in software production, *IFIP Transactions* 8(2): 315–330.
- Trienekens, J.J.M. and van der Zwan, M. 1997. Service Level Specificatie, Automatiseringsgids (in Dutch), pp. 23–24.
- Trienekens, J.J.M. and van Veenendaal, E. 1997. *Software Quality from a Business Perspective, Directions and Advanced Approaches*. Kluwer Bedrijfsinformatie.



Jos J.M. Trienekens is an Associate Professor at TU Eindhoven (University of Technology—Eindhoven) in the area of ICT systems development. He is responsible for a research program on ICT driven business performance and is an associate member of the research school BETA at TUE that focuses at operations management issues. Jos Trienekens published over the last ten years various books, and papers in international journals and conference proceedings. He joined several international conferences as PC member and member of the organization committee. He also has experience as project partner in several European projects (ESPRIT/ESSI) and is on a part-time basis with KEMA Quality, a service organization and certification body in The Netherlands.



Jacques J. Bouman graduated from the Eindhoven University of Technology at the Department of Computer Science. Since then he has been working at the same University at the Department of Technology Management where he has done research in the field of IT Service Management. At the moment his research is focusing on E-learning, specifically the field of Distributed Collaborative Learning.



Mark van der Zwan is a senior consultant at Improve Quality Services, a business consultancy and service organization in The Netherlands. He graduated from Eindhoven University of Technology in 1995. He is currently involved in consultancy and research projects in the area of service management, test process improvement and software quality evaluation.