# Specifying business services: learning from software engineering

Joaquin Peña-Siles
*Department of Computer Systems and Languages, University of Seville, Seville, Spain, and*

Maria del Mar González-Zamora and José A.D. Machuca
*Department of Financial Economics and Operations Management, University of Seville, Seville, Spain*

Received May 2009
Revised 30 June 2010, 24 February 2011
Accepted 24 August 2011

## Abstract

**Purpose** – The specification of business services (BS) is one of the key factors for success in service provision. Researchers and practitioners have identified a set of problems in BS specification, namely: communication problems between providers and buyers, inaccurate specifications and changes in requirements. These problems were identified in the software engineering (SE) field many years ago, resulting in the development of many techniques and tools to address them. Given the similarities between the two fields, the purpose of this paper is to identify the main lessons learned in SE and to propose how they can be adapted to the BS field.

**Design/methodology/approach** – The paper takes the form of a literature study and presents a conceptual solution.

**Findings** – To address communications problems and inaccuracies in BS specification, a formal requirements specification stage is required. Requirements should be set out in a well-structured written, definition, document which can be used at a subsequent stage for design of the service (the definition of the solution). A requirements document for BS is proposed based on BS literature and SE methods. This document is based on the approaches used in the SE field for improving communication and reducing inaccuracies and covers the information needed in the BS field for specifying a business service. Second, a life cycle approach is proposed based on SE practice. It is shown that different lifecycle sequences can be used depending on the degree of fluidity in the communication between buyer and provider, the complexity of the BS and the stability of their requirements, with a repetitive process where specification and design are revisited iteratively and/or incrementally often being the most appropriate. This approach helps to stabilise requirements and to avoid inaccuracies being made in the specification.

**Research limitations/implications** – This paper is literature based. Although still untested empirically in the BS field, the recommended approach has been intensively proved in the SE field.

**Originality/value** – Requirements specification is a necessary (monetary, time and resource) cost for successful BS provision. The BS industry must realise that requirements have to be set down in writing and agreed upon with customers before initiating the design of the service.

**Keywords** Software engineering, Customer requirements, Specifications, Business planning, Business services

**Paper type** Conceptual paper

## 1. Introduction

Specification and design of Business Services (BS) has been seen as one of the keys to successfully developing and selling BS (Smeltzer and Ogden, 2002; Andersson and Norrman, 2002, Jackson *et al.* (1995); van der Valk and Rozemeijer, 2009). Despite

Journal of Service Management
Vol. 23 No. 1, 2012
pp. 97-119

1757-5818
DOI 10.1108/09564231211208998

their importance, developing written specifications for the service desired remains a challenge (Fitzsimmons and Fitzsimmons, 2008) and arriving at clear specifications is no easy task. Smeltzer and Ogden (2002) argue that service specification is the second most complex task in the purchasing process; and many purchasing companies may seem reluctant to address this part of the purchase process thoroughly (van der Valk and Rozemeijer, 2009).

It has been argued that there are three main problems that cause this complexity (Åhlström and Nordin, 2006; Ellram *et al.*, 2008):

(1) *Communication problems between service provider and service buyer.* To provide a successful service, providers should have an in-depth understanding of the buyers' problems, and buyers should have an in-depth understanding of the range of solutions that providers can offer. This is not easy, since the culture, area of expertise, vocabulary, etc. of buyers and providers are often very different, making communication difficult.

(2) *Inaccurate specifications.* A proper description of the service is invaluable. Given the communication problems, specifications should be set down carefully in writing to ensure that buyer and provider understand each other and agree on all relevant details.

(3) *Changes in service requirements.* The services offered must be competitive and improved continuously. Buyers do not usually understand what the provider can offer them nor explicitly know their own needs leading to a process where the buyer and the provider come to understand each other step-by-step. Thus, with each step forward buyers gain a better understanding of what they need and what can be obtained from the provider, changing/adding/deleting requirements from previous steps.

In the services management field there are models, methods and tools devoted to BS design, for example quality function deployment, structured analysis and design technique, failure mode and effects analysis, service blueprinting and reference models (Bullinger *et al.*, 2003). However, none of these addresses the above problems as they do not pay enough attention to the specification phase. They all tend to assume that all the buyer's requirements are stable and clearly specified and proceed directly to the service definition.

The software industry has had to contend with the same three problems mentioned previously (Standish Group, 1995). As a result, in recent decades many approaches and solutions have been presented and adopted by the industry. The main conclusions in the software industry are that a great deal of attention should be paid to specifying requirements and to setting up a variety of well-structured stages to reduce the impact of these problems (Jacobson *et al.*, 1993; Cockburn, 1997, 2000; Gottesdiener, 2002; Robertson and Robertson, 2006).

We can consider the development and provision of software as a provision of a BS. Thus, the software industry can be seen as a specific case of the BS industry where BS are sold. This paper takes these similarities as its basis to identify the main lessons learned in software engineering (SE) and adapt them to the BS field.

The remainder of this paper is organised as follows: the next section explores problems common to the SE and BS fields in greater detail and shows how these problems have been addressed by SE. Section 3 proposes a structure for BS specification analysis the information to be covered. The detail of this structure is presented in the Appendix, using the example of a Hotel Chain which outsources a service for selling its rooms online. Section 4 presents a lifecycle approach. Finally, Section 5 shows the main conclusions and opportunities for future research.

## 2. Addressing problems common to BS and SE

Building software is a very complex task as it comprises a large set of instructions that must be executed by a machine. For example, the Windows XP operating system contained 40 million of these instructions all of which have been written by people who have to coordinate in order to create a service that exactly meets the needs of a wide variety of service consumers (users). The term "software crisis" was coined at the first NATO Software Engineering Conference in 1968. The term reflected the main challenge that the software industry faced from its beginnings: how to organise, define and align such a huge number of small steps to meet customer's needs given that these needs evolve over time.

A study by the Standish Group (1995), titled "The Chaos Report" led to an inflection point in the industry. It found that most major software projects failed and that the projects that were completed took twice the time and cost twice as much as had been originally estimated. The report also identified a number of major problems very similar to those in the BS industry described earlier. The three main problems were:

(1) Lack of user input due to communication problems (12.8 per cent).

(2) Incomplete requirements and specifications (12.3 per cent).

(3) Changing requirements and specifications (11.8 per cent).

Prior to this, little attention had been paid to requirements specifications in the SE field. Software specifications had normally started with technical models not usually understood by customers. Models of this kind were an obstacle to requirements development, and led to a large number of inaccuracies being made as a result of misunderstandings between buyers and providers. This led to the recognition of the need for:

- Detailed and organised textual descriptions of the requirements in form understandable by the customer – the reason for the services – and detailed descriptions of solutions to the requirements.
- Mechanisms to help customers clarify what their requirements are exactly and what is the solution that best suits their needs.

This in turn led to the development of the discipline of requirements engineering focusing on providing the techniques, professionals and methodologies needed to elicit and document customers' needs using textual descriptions.

As in the early days of the SE industry, there is evidence that in the BS field little attention had been paid to service specification (Johnston, 2005; Pinto and Johnston, 2009). One reason being that, despite being crucial for building a service that meets the buyer's needs (Fitzsimmons and Fitzsimmons, 2004; Matthyssens and Vandenbempt, 1998; Ojasalo, 2007), specification was seen as an unnecessary cost due to the impression that services are simpler than they really are (Congram and Epelman, 1995). However, it has been argued that in the BS field there are similar problems to those found in the Chaos Report (Åhlström and Nordin, 2006; Ellram *et al.*, 2008), as was stated in Section 1, namely:

- Communication problems between the service provider and the service buyer.
- Inaccurate specifications/requirements.
- Service requirement changes.

In the following sections, we detail each of these problems and outline the main solutions adopted in the SE field.

*2.1 Communication problems*

A major problem identified in the SE industry has been communication between the software buyer and the software builder. Before the Chaos Report, it was assumed by most of the industry that customers knew what their needs were, but it was subsequently realised that:

- Customers do not usually know what software can offer them and they do not have the sufficiently detailed description of their own problem in their minds that is required for building software.
- The software company is not necessarily an expert in the domain of the customer's problem and does not usually understand it.

We argue that the same problems occur in BS where a service is often so complex that it is very difficult for buyers to state exactly what they need (Fitzsimmons and Fitzsimmons, 2004; Matthyssens and Vandenbempt, 1998; Ojasalo, 2007). This situation is aggravated by buyers not usually knowing/understanding what the provider can offer them. Another problem identified in the BS field is that when buyers describe what they want, it is not necessarily the service that they really need. As in the SE field, service providers apparently need to have excellent diagnostic abilities in order to discover what the buyer really needs.

The following approaches have been adopted in the SE field to address communication problems:

- Every project should start with a "Requirements elicitation" stage, with the purpose of creating a text-based document in which the customer and the provider can agree on the description of the service required or the problem.
- A glossary of definitions of technical terms should be added to requirements documents. This reduces communication problems since the specific terms used by buyer and provider in the specification are defined (Gottesdiener, 2002).
- The document must model the customer's business processes. This improves the supplier's understanding of the customer, reducing communication problems.
- This document should be subsequently analysed. This analysis process is based on translating textual descriptions (requirements document) into unambiguous technical diagrams and set out in an analysis document[1]. The purpose of this document is twofold: first, to detect errors, inconsistencies, duplicate information and ambiguities in the textual requirements document; and second, to improve and detail the description of the problem. Both, requirements and analysis documents can continually provide each other with feedback until the requirements document is free of errors and ambiguities and is finally validated by the customer.

We propose that the above approaches can readily be adapted to a BS context; with both the buyer and the provider be involved in specifying the service.

*2.2 Inaccuracies in service specifications*

This is a problem in both the SE and BS fields. In the BS field, a full and clear description of the service is one of the key factors for providing the required service (Andersson and Norrman, 2002; Bullinger *et al.*, 2003). Under-specification is common and has been identified as one of the most important problems facing BS (Åhlström and Nordin, 2006; Ellram *et al.*, 2008).

This has been addressed in the SE field; requirements need to be explicitly documented before starting the design of the service. Software engineers use three main tools to achieve unambiguous requirements specifications (Jacobson *et al.*, 1993; Cockburn, 2000; Durán, 2000):

- *List of items to be covered.* This provides well-structured documents and ensures that the specification will be complete since there is a list of information that must be complied with.
- *Tabular templates.* This technique is used to avoid incomplete information being provided. They contain a list of attributes that must be covered to define each requirement, ensuring that the specification is comprehensively met. It also improves readability and understanding since the information is always presented in the same format.
- *Sentence patterns.* This technique consists of using predefined parts of the sentences used in the tabular templates to improve readability and understanding and avoid any ambiguity in the requirements description that might result from free descriptions.

In a written document, it is also common for many parts to be interrelated and for there to be a large amount of information. The analysis phase checks the textual information for inaccuracies and incompleteness. These approaches are important for BS as specification techniques in the BS area focus on a technical description of the service, not the requirements documentation.

*2.3 Changes in service requirements*

This again is a problem impacting both SE and BS. In BS, it has been observed that requirements change throughout the service specification process should also evolve over time to adapt to changes in the market and to maintain any competitive advantage (Ellram *et al.*, 2008; Bullinger *et al.*, 2003). In the SE field, changes in users' evolving needs were one of the factors that contributed to project failure identified in the Chaos Report and a number of approaches and tools have been developed to address managing changes:

- *Documenting requirements stability, importance and urgency.* Attributes can be used to score the stability, importance and urgency of each requirement. This in turn allows requirements to be ranked and those with the highest scores to be designed/implemented first.
- *Incremental and iterative processes.* These are well-suited to coping with unstable requirements. Ideally there should be short iterations where all the phases

of software development are covered and culminate in a small part of the software being completed. In this way, customers can see part of the solution and, with a better understanding of their needs, fine tune the rest of requirements. Requirements are, therefore, rethought after each iteration with implementation of those with higher scores for stability, importance and urgency. This allows the solution to the customer's needs to be easily optimized by managing changes in a very agile way.

- *Traceability*. This consists of documenting dependencies between requirements. For example, the interactions between buyer and seller can be documented indicating the dependencies each interaction has with other elements, such as items exchanged in the interaction, roles involved, or goals pursued by service purchasers, etc. Using this, information matrixes can be constructed to illustrate these relationships, which allow the impact of a change to be evaluated. It also shows, for example, which items need to be reviewed when some other item is changed.

In this paper, we argue that a similar approach can be valuable for BS. Thus, as in the SE field, to address the three problems described, a set of processes and a structure for specification can be used. In the following sections, we explore how to structure and develop a BS specification. The following section presents the specification structure.

## 3. Developing BS specifications

We have argued in the previous section that to address these problems it is vital that requirements are properly identified and documented. The information covered in the specification must be complete and consistent. In this section, we argue that the information to be considered in BS is similar to that in SE and outline the key areas to be covered.

### *3.1 Information required*

In SE, there is a well-known set of items that should be covered in any requirements document (Durán, 2000; Cockburn, 1997; Sommerville and Sawyer, 1997; Robertson and Robertson, 2006). However, there is no such set of items for BS requirements. Bullinger *et al.* (2003), based on Donabedian (1980), has proposed three part classification for BS:

(1) Resource model (structure dimension): the resources required for the service to be provided correctly, comprising human resource capabilities, such as the definition of the skills required by the staff involved, information systems required, etc.

(2) Process model (process dimension): describes how the outcomes of the service are achieved.

(3) Product model (outcome dimension): describes what a service does and its outcomes (tangible and intangible).

We use this classification to explore how the lessons from SE requirements can be adapted for BS.

*3.1.1 Requirements for the resource model*. We propose that the information needed for the BS resource model should include:

- *Functional requirements based on interactions between organisations.* Specifications should take into account that BS are produced and consumed in interactive processes between the buyer and the provider. This requires specifying the interactions between the buyer and provider during the service exchange. This specification should contain a description of the interactions between employees from both sides of the relationship, the relationships at each level of the hierarchy and the different functions involved in every organisations (Åhlström and Nordin, 2006; Fitzsimmons and Fitzsimmons, 2004; Wynstra *et al.*, 2006). In the SE field, the functional requirements are called the use case; this describes the interactions between users and the software. It uses a tabular template which shows interaction as a number of steps described in plain text. These steps describe the actions performed by a user playing a role and how the software reacts. Although use cases focus on the interactions, they also describe the participants in these interactions in the form of roles (see below). From the resources point of view, use cases provide information on the functional organisation needed for performing the service.
  Many BS companies seem to have neglected this aspect of ongoing interaction. Buyers that define and design their daily dealings with the supplier (in terms of people involved, communication channels and topics to be discussed), experience fewer problems during the contract period. In addition, buyers that detail the relationship with suppliers are relatively more satisfied than those who do not detail the relationship, with both the process of service delivery and the service delivered (van der Valk and Rozemeijer, 2009). Designing the ongoing interactions is something that has to take place in advance of signing the contract, that is, in the initial stages of the buying process (van der Valk and Rozemeijer, 2009).
- *Organisational structure.* In the SE field, roles participating in use cases are specified. These are all the users of the software system and their relationships. Note that in SE the organisational structure, human resources and resources needed for the software to be used have not been regarded as problems. In BS, the roles of both the provider and the buyer are important for specifying these interactions (Roth and Menor, 2003; Bullinger *et al.*, 2003). Indeed, an unclear or incomplete description of the responsibilities and roles involved has been identified as one of the problems in establishing service supply relationships (Kelly *et al.*, 2002; Åhlström and Nordin, 2006).

We thus propose that BS requirements should include specification of functional requirements or use cases, and roles.

*3.1.2 Requirements for the process model.* We propose that the following information is needed for BS with regard to the process model:

(1) *Business processes.* The provider needs to have a detailed understanding of the customer's business processes, in order to be able to offer the most suitable service solution that fits their needs and way of operating (Ojasalo, 2007; van der Valk and Rozemeijer, 2009). The information used in the SE field is similar to that proposed by different authors in the BS field. Thus, we propose to adapt the business process models used in SE field. These show the software buyer's operations in order to identify which of these are to be outsourced

to a software system and to better define and understand the buyer's requirements. In the SE field, the most common techniques used to model business processes are business process modelling notation (BPMN) and activity diagrams (Object Management Group, 2009).

(2) *Interactions between business processes.* Building on the resource model; for a service to be provided the buyer and the provider should interact in a business process. These interactions should be described carefully in order to synchronise and integrate both organisations' business processes (van der Valk and Rozemeijer, 2009). The SE field addresses this through:
   - *Use cases*: as mentioned previously, these describe how the software and users interact. Thus, use cases focus on interactions while business processes focus on user operations. In general, terms, business processes provide the general context of the operations, while use cases define internally those steps of the business process that involve the use of software. Thus, they are at a finer grained specification level than business processes.
   - *Use case diagrams*: these diagrams show graphically which roles interact between each other via use cases.
   - *System goals*: in the SE field, these specify the functional goals that the software system must achieve. They complement the information provided in the business process by showing the purpose that is pursued by them. In BS, these goals can be used to represent strategic, tactical and operational goals related to each participant's BS and their alignment, since alignment has been presented as a prerequisite for forging successful and stable commercial relationships (Lee, 2004; Åhlström and Nordin, 2006).

*3.1.3 Requirements for the product model.* We propose the following information to be provided for the BS product model:

- *Items exchanged between participants.* Since almost all services are a mixture of tangible and intangible components, BS specifications must also cover the features of the goods (tangible components) exchanged (Fitzsimmons and Fitzsimmons, 2008). In the SE field, specifically in information systems, it is necessary to specify the items to be exchanged. Since in the SE field the inputs and outputs of the system are data, such items are the information to be exchanged between the software system and its users. This is covered in the SE requirements document through the information requirements. We propose that this should be adapted for use in the BS field for representing the goods exchanged. As in SE, where the features of the information to be exchanged must be detailed, in the BS field the characteristics of each kind of good exchanged in the service should be specified.
- *Non-functional requirements.* As mentioned above, almost all services are a mixture of tangible and intangible components. Thus, BS specifications must also cover its intangible features. These are requirement that specify criteria that can be used to measure/specify the quality that must be provided during the execution of the service (represented previously as business processes). This should be contrasted with functional requirements that define the operational steps to be performed. Thus, an example of non-functional requirements for

a service can be the security level provided, the maximum time for replying to a request, or a 90 per cent customer satisfaction level. These requirements are of major importance in SE, as indeed they are in BS (Matthyssens and Vandenbempt, 1998). Thus, quality of service, often embodied in a service level agreement, is an important component of the information needed in the BS requirement for the product model. The dynamics of the relationship between customer and supplier are an integral part of the quality of the service (QoS). The process of evaluating quality is differs between services and goods. Something that is tangible can be inspected and examined over time, but as services are for the most part intangible, they cannot. In addition, the assessment of quality is made during the service delivery process. Therefore, collaboration between the supplier and the buyer is a way of ensuring the delivery of a quality service, and given the intangible nature of industrial services, the challenge is to explicitly communicate the quality that service providers offer (Matthyssens and Vandenbempt, 1998; Ellram *et al.*, 2004).

*3.2 A BS specification document structure*

Given that service buyers do not have to be experts in modelling techniques and methodologies for specifying services, and given that communication between provider and buyer needs to be unambiguous and problem-free, we argue that as in SE, documents written in natural language should be used (Cockburn, 1997). Thus, we propose that BS specifications should follow a similar approach to those used in the SE field for writing and structuring the requirements documents, and contain a similar set of information. As in SE, we believe that BS documents should be well structured. It would be useful to have requirements specialists to cover all the aspects that are relevant for service specification. This improves the chances of arriving at a more comprehensive specification and avoiding inaccurate specifications.

Building on our previous discussion and based mainly on Durán (2000), we propose an adaptation of SE specification document structure in a document structure for BS specifications. This structure is outlined in Table I. The first two columns outline the document structure. The next two columns of this table show the information required in BS and its equivalence with the required information in SE. A hypothetical example of a requirements document is shown in the Appendix.

*3.2.1 Developing the document*. Earlier we highlighted a number of potential problems in the development of BS specifications. We propose that in developing a specification document we can build on and adapt some of the techniques used in SE.

Avoiding inaccurate specifications. There are three main techniques and tools that can be used. First, a clear and complete index (list of sections) should be used to avoid incompleteness. Second, a set of templates for specifying the items in each section can be used to avoid inaccuracies and incompleteness (Durán, 2000; Cockburn, 1997; Sommerville and Sawyer, 1997; Robertson and Robertson, 2006). A final technique is to use sentence patterns to avoid free writing and consequent ambiguity (Durán, 2000).

Examples, based on a hotel example, of the templates used in the main sections (4-6) of the proposed document are outlined in the Appendix. Each template is presented with an example (hotels and online reservations) and is annotated to explain the purpose of each attribute. Note that sentence patterns are used in all templates; italics indicate manually-added text and normal Roman font, fixed parts of the sentence.

**Table I.**
Proposed structure for BS requirements documents

| Section | Description | Aspects to be covered in BS specifications | Equivalent in software engineering |
|---|---|---|---|
| 1. Introduction | Introduction to the enterprise that is to provide the service and to the service | | Introduction |
| 2. Participants in the service specification | Lists the organisations and people involved in writing the requirements specification | Organisation structure and responsibilities of roles | Participants in the specification |
| 2.1. Organisation X | Introduction to organisation X and how it participates in the specification<br>List of templates showing people involved in the specification | | Description of organisations |
| 3. Current business processes | Introduction to buyer's current process for the service to be purchased and also future operations after the service has been used | Business processes and interactions between business processes | Business process models |
| 4. Service strategy and goals | Introduction to participants' strategies and goals | | System goals |
| 4.1. Strategies and goals | Lists the strategies and goals to be achieved by an organisation<br>List of strategy and goal templates | | System goals |
| 4.2. Alignment: strategy and goals that align participants' interests | List of strategy and goal alignment templates: written description of alignment strategies and goals that participant organisations share | Interactions between organisations and business processes | System goals |
| 5. Requirements catalogue | Lists the service requirements | | |
| 5.1. Exchanged items requirements | List of templates showing items exchanged between participants in the service | Items exchanged between participants | Information requirements |
| 5.2. Functional requirements | Contains the functional requirements | | Functional requirements |
| 5.2.1. Roles in the service | Lists participants in the service by business unit (department)<br>List of role templates showing all participants in the service by org. and dept. | Organisational structure and responsibilities of roles | Roles participating in use cases |
| 5.2.2. Interactions | Lists interactions between participants | Interactions between organisations | Use cases |

(*continued*)

| Section | Description | Aspects to be covered in BS specifications | Equivalent in software engineering |
|---|---|---|---|
| 5.2.2.1. Interaction diagrams | | | |
| | List of diagrams showing interactions between role participants in the service | Interactions between organisations | Use case diagrams |
| 5.2.2.2. Interaction cases | List of interaction case templates showing interactions between roles in the previous section ordered by subservice | Interactions between organisations | List of use cases |
| 5.3. Non-functional requirements | Lists the non-functional requirements | | Non-functional requirements |
| 5.3.1. Quality of service | Lists QoS requirements | Specification of the quality of service | Non-functional requirements |
| 5.3.2. Legal, moral and ethical aspects | Lists the legal, moral and ethical constraints that are applicable | | |
| 5.3.3. Cultural requirements | Lists the cultural requirements for the service to be provided successfully | | |
| 6. Traceability matrixes | Shows requirements dependencies | | Traceability matrixes |
| 7. Glossary of terms | Lists definitions of specific terms in the domain that may make communication difficult | | Glossary of terms |
| 8. Meetings | Contains transcripts and summaries of meetings | | Meetings |
| 9. Documentation | Contains documentation provided by each participant as inputs for the specification | | Documentation |

**Table I.**

Managing changes in requirements. There are three tools that can be used to support this in requirements documents.

First, all the templates used for documenting each item of the index should present a score for its *Importance, Urgency and Stability* since these attributes allow changes to be managed. When the requirements have been ranked with a score for each attribute, the most important, urgent and stable items can be covered first. Once these items have been used to design part of the service, the customer can reconsider the remaining, less stable requirements, redefine them and assign new scores to their attributes. These processes are executed in the software field using iterative/incremental lifecycles.

Second, for managing changes it is also important for the names of the people involved in the specification to be recorded on each template. Thus, when addressing changes, conflicts, inconsistencies and mistakes, the right people can be contacted to address the conflicts. Whether the requirement has been validated by the buyers, contains errors, is a draft, has been implemented, etc. should also be indicated.

Third, all items must also be given an ID so that dependencies with other items can be established. These dependencies are used to build the traceability matrixes (See Section 7 of the Appendix) and to assess whether all the goals have been covered by at least one exchanged item requirement, one interaction requirement and one quality of service template, or to evaluate the impact of a change.

Addressing potential communication problems. We propose the following.

First, in order to improve communication and understanding, the customer's current and future operations (after the service has been deployed) should be specified (see Section 3 of the requirements document). This helps the service provider to understand the buyer's problems and to clarify which parts of the process will be outsourced/externalised, which also provides an initial definition of the problem to be covered.

Second, we propose that two techniques that are used in SE for documenting current business processes are valid for BS. The first is use case templates. An example is shown in Section 5 of the Appendix, the second is business process model such as flow charts, BPMN or activity diagrams (Object Management Group, 2009). Business process models are recommended when the buyer provides experts who can manage them to take part in the service specification. If this is not possible, the use of use case templates is recommended as they are written descriptions that can be easily understood by non-experts.

Finally, we propose that there should be a glossary of terms containing technical vocabulary used by the buyer that is not commonly known by the seller, and vice versa.

## 4. A lifecycle approach

The development of a specification follows a life cycle. The lifecycle used in SE is shown in Figure 1. The first two stages are the most important for the goal of this paper, providing solutions to communication and inaccuracy problems. The first stage is Requirements Elicitation. It focuses on documenting customer needs in the "requirements document", which is used as a communication tool where buyer and provider can agree on the features of the software. Next, the analysis phase focuses on transforming the written document into "formal" models that are analysed in order to detect any inaccuracies, conflicts, duplicate information, etc. in the written document. This stage also uses tools that allow a detailed understanding of buyer's requirements to be had and for formally defining the problems which are to be used in the following stages.

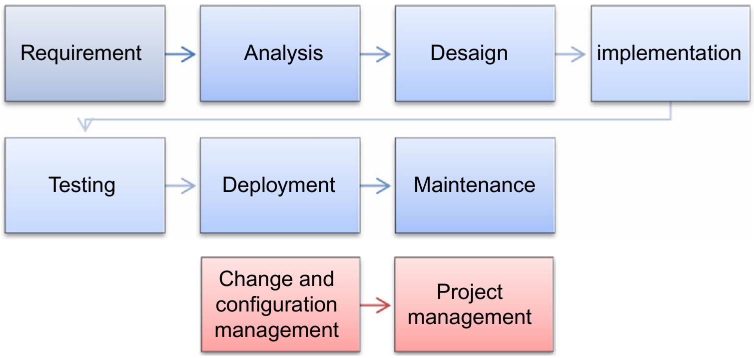



**Figure 1.** Software engineering lifecycle

Following the requirements and analysis stages, in the SE field there are a set of further stages that follow the development of a successful specification. These stages consist of designing the solution (Design), implementing it on the basis of the analysis models (Implementation), detecting codifying errors (Testing), installing the software on the client's IT infrastructure (Deployment) and keeping the software running properly (Maintenance).

There are also two activities that should be performed in parallel for every software project. First managing all the versions and configurations of the software and the requests for changes made by customers (change and configuration management). Second, project management, covers all the tasks that ensure that the project is finished on time and within the resource estimate.

This life-cycle approach can also be used in the services field as a framework for providing structure (Bullinger *et al.*, 2003). A BS lifecycle can be defined as a framework containing the processes, activities and tasks involved in the definition, development, operation and maintenance of BS, spanning the life of the service from its definition until the end of its use (ISO/IEC, 12207).

The lifecycle phases can be redefined for use in the BS field:

(1) *Requirements elicitation and specification.* During this phase the buyer's needs are set out in a service specification.

(2) *Analysis.* This analyses the buyer's needs in order to detect inaccuracies and incompleteness and to obtain a well-structured description of the buyer's problem.

(3) *Design.* The main focus of design is to describe how the buyer's problem is to be solved by service provider capabilities.

(4) *Implementation.* This consists of deploying the service and beginning to provide it. The main tasks in this stage are:

- testing the service in its final execution environment;
- providing assistance and support to service users; and
- educating users, suppliers and customers, in the proper use of the service.

(5) *Provision, maintenance and testing.* this phase consists of maintaining the service ensuring quality and checking for/solving errors. In particular, this phase identifies and documents any defects in the customer perceived

service quality. It also provides validation of the assumptions made in the design and specification of requirements with concrete evidence, and ensures that actual service features are compliant with those designed.

(6) *Project management.* Manages goals, risks, constraints and resources assigned to the project. It also ensures the proper development of the project and verifies that all the milestones are reached correctly.

Managers have choices in the actual application of the life cycle. The most basic lifecycle consists of executing all phases in succession. Thus, the outcomes of upstream phases provide the input for downstream phases. This is known as the waterfall schema. Subdividing the process into predefined steps makes this schema very transparent but at the expense of flexibility (Bullinger *et al.*, 2003). As the process follows a rigid pattern, it leaving little scope for adaptation to a special service, this schema is appropriate for standardised, low risk, low variability and low evolution services.

As Zomerdijk and Voss (2011) point out, many services seem to be in "perpetual Beta". Thus, a single specification development lifecycle would be inappropriate. There are two possible models which could be much more effective in many services:

(1) An iterative schema covers all the phases repeatedly until the service is finished (Figure 2). Each new version represents the complete service with improvements over the previous version. It is rare as a method for developing services (Bullinger *et al.*, 2003). The advantage of this schema is that the service is available to users from the very beginning, allowing them to propose improvements during each iteration and enabling errors to be corrected in the subsequent cycles. Because of its complexity, the use of this method is advisable for services that affect core operations, those whose impact on the organisation is high, or those whose implementation is time-consuming.

(2) Incremental schema consists of performing a waterfall lifecycle repeatedly until arriving at a finished version of the service (Figure 3). However, unlike in the incremental lifecycle, a set of new features can be added during each iteration. Each iteration provides a finished version of part of the service. This method is useful when there are unstable requisites.

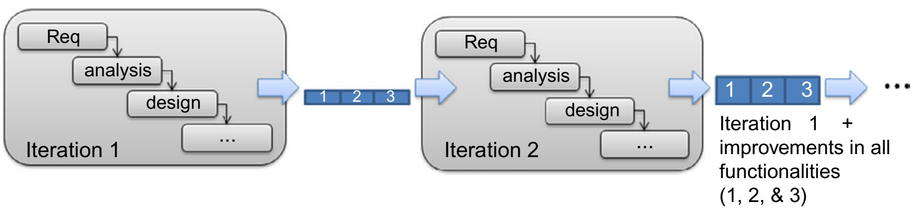



**Figure 2.**
Iterative schema

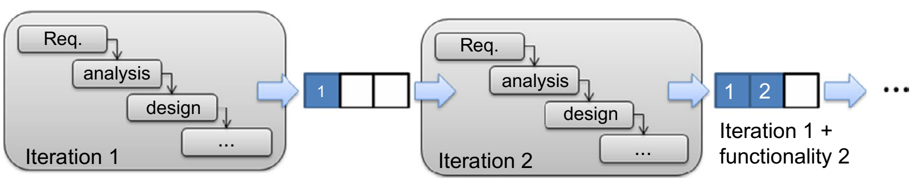



**Figure 3.**
Incremental schema

A second managerial choice is the amount of documentation and number of procedures used. Lightweight processes are those that put a low load on procedures and use as little documentation as possible. This is well-suited to small/medium-size organisations that do not have enough human resources to execute highly bureaucratic methodologies. It is also appropriate for service provisions where the requirements are unstable since documenting frequently changing requirements introduces an unnecessary overload and slows down project development. Heavyweight processes require the service to be highly documented with highly bureaucratic procedures. This is more appropriate in service provision where documentation is essential for orchestrating and communicating the large number of actors present in big organisations. In addition, heavyweight processes work well for services where requirements are stable or involve core services where mistakes or inaccuracies have a profound impact on the organisation.

These managerial choices and the conditions under which they are appropriate are shown in Table II. The choice of processes used in SE field depends on how fluid the communication between buyer and provider is and how stable the requirements are (Cockburn, 2008; Larman and Basili, 2003). Given the similarities between SE and BS field, we propose that this is equally applicable in BS.

## 5. Conclusions and future research

The development of specifications has been identified as an important gap in BS (Johnston, 2005) with no formal tools or processes to carry it out. It is also a complex task due, in particular, to the three main problems identified in this paper; communication problems between service provider and service buyer, inaccurate specifications and changes in service requirements. The software industry has had to face the same three problems and developed processes and tools for software specification to address these issues. We argue that these adapt well to the BS industry due to the similarities between the two fields.

To address the development of BS specifications, this paper first proposes that a lifecycle approach including a requirements specification phase should part of a BS development process. Its purpose is to set out requirements in a well-structured written document in which the buyer and the supplier can agree on the specification of the service. Building on SE methods, a requirements document for BS is proposed.

The potential benefits to the BS industry can far outweigh the costs of such a disciplined and structured approach and should be seen as a for successful service provision. We believe that the approach presented in this paper can be applied to any type of service but, as is the case with SE, the more complex the service to be specified and the more unstable the requirements are, the more useful it will be.

| Communication | Changes in service requirements | |
|---|---|---|
| Very fluid | Lightweight processes<br>Incremental schema | Lightweight processes<br>Iterative schema<br>(Waterfall schema in case of non-complex services) |
| Not very fluid | Heavyweight processes<br>Incremental schema | Heavyweight processes<br>Iterative schema |
| | Unstable | Stable |

**Table II.**
Guide to selecting processes

This paper also puts forward the choices that management have in sequencing BS development stages. Iterative/incremental processes can be used to reduce inaccuracies and to address changing requirements, and repeated lifecycles, where requirements and analysis are successively revisited, are the most suitable for complex and unstable requirements.

The main limitation of this research is that, although conceptually and theoretically sound, it has not been subject to major empirical testing. We, therefore, see two possible directions for future research. The first is to build on the analysis in this paper and to develop analysis models that help:

(1) To detect inaccuracies, errors and redundancies in requirements documents.

(2) To perform a deeper analysis of the buyer's problem, for example development of an analysis document to support the proposed requirements document (Bocanegra *et al.*, 2009).

The second is empirical testing of the proposed models, for example assessment of the attributes included in requirements templates by means of an empirical study conducted in a number of companies in the service industry.

## Acknowledgements

The authors thank the Associate Editor and anonymous referees for their valuable feedback, which clearly helped to improve this paper. This study was conducted within the framework of the Spanish Ministry of Education and Science National Programme of Industrial Design and Production (DPI-2006-05531 and DPI-2009-11148) and the Junta de Andalucía (Spain) PAIDI (Plan Andaluz de Investigación, Desarrollo e Innovación) Excellence Projects (P08-SEJ-03841). This work has been partially supported by the European Commission (FEDER) and the Spanish Government under the CICYT project SETI (TIN2009-07366), and by the Andalusian Government under the projects ISABEL (P07-TIC-2533 and TIC-5906) and THEOS (TIC-5906).

## Note

1. These diagrams represent the textual description in an unambiguous language which allows the detection of inaccuracies in the text and their correction. An example of these diagrams is the "class diagram", which are used to represent information exchanged. The class diagrams have an underlying formal theory that defines which are the types of relationships that can be found between information in an unambiguous way. For example, two information units (called "classes") can relate by "aggregation". That means that one of the information units is the container of the other one (e.g. a book is an "aggregation" of pages).

## References

Åhlström, P. and Nordin, F. (2006), "Problems of establishing service supply relationships: evidence from a high-tech manufacturing company", *Journal of Purchasing and Supply Management*, Vol. 12 No. 2, pp. 75-89.

Andersson, D. and Norrman, A. (2002), "Procurement of logistics services-a minutes work or a multiple year project?", *European Journal of Purchasing & Supply Management*, Vol. 8 No. 1, pp. 3-14.

Bocanegra, J., Peña, J. and Ruiz-Cortés, A. (2009), "Interorganizational business modelling: an approach for maintaining traceability between goals, processes and organizational structures", *Iberoamerican Conference on Requirement Engineering and Software* (*IDEAS 09*), *p. 15*.

Bullinger, H.J., Fähnrich, K.P. and Meiren, T. (2003), "Service engineering-methodical development of new service products", *International Journal of Production Economics*, Vol. 85 No. 3, pp. 275-87.

Cockburn, A. (1997), "Structuring use cases with goals", *Journal of Object-oriented Programming*, Vol. 9 No. 5, pp. 35-40 (Vol. 9 No. 6, pp. 56-62).

Cockburn, A. (2000), *Writing Effective Use Cases*, Addison-Wesley, Reading, MA.

Cockburn, A. (2008), "Using both incremental and iterative development", *STSC CrossTalk* (*USAF Software Technology Support Center*), Vol. 21 No. 5, pp. 27-30.

Congram, C. and Epelman, M. (1995), "How to describe your service: an invitation to the structured analysis and design technique", *International Journal of Service Industry Management*, Vol. 6 No. 2, pp. 6-23.

Donabedian, A. (1980), "Explorations in quality assessment and monitoring", *Volume I: The Definition of Quality and Approaches to Its Assessment*, Health Administration Press, Ann Arbor, MI.

Durán, A. (2000), "A methodological environment for requirement engineering of information systems", PhD thesis, University of Seville, Seville.

Ellram, L.M., Tate, W.L. and Billington, C. (2004), "Understanding and managing the service supply chain", *Journal of Supply Chain Management*, Vol. 40 No. 4, pp. 17-32.

Ellram, L.M., Tate, W.L. and Billington, C. (2008), "Offshore outsourcing of professional services: a transaction cost economics perspective", *Journal of Operations Management*, Vol. 26 No. 2, pp. 148-63.

Fitzsimmons, J.A. and Fitzsimmons, M.J. (2004), *Service Management: Operations, Strategy and Information Technology*, MGraw-Hill, New York, NY.

Fitzsimmons, J.A. and Fitzsimmons, M.J. (2008), *Service Management: Operations, Strategy and Information Technology*, MGraw-Hill, New York, NY.

Gottesdiener, E. (2002), *Requirements by Collaboration: Workshops for Defining Needs*, Addison-Wesley, Reading, MA.

ISO/IEC 12207 (n.d.), Information Technology/Software Life Cycle Processes.

Jackson, R.W., Neidell, L.A. and Lunsford, D.A. (1995), "An empirical investigation of the differences in goods and services as perceived by organizational buyers", *Industrial Marketing Management*, Vol. 24 No. 2, pp. 99-108.

Jacobson, I., Christerson, M., Jonsson, P. and Övergaard, G. (1993), *Object-oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, Reading, MA.

Johnston, R. (2005), "Service operations management: from the roots up", *International Journal of Operations & Production Management*, Vol. 25 No. 12, pp. 1298-308.

Kelly, M.J., Schaan, J.L. and Joncas, H. (2002), "Managing alliance relationships: key challenges in the early stages of collaboration", *R&D Management*, Vol. 32 No. 1, pp. 11-22.

Larman, C. and Basili, V.R. (2003), "Iterative and incremental development: a brief history", *IEEE Computer*, Vol. 36 No. 6, pp. 47-56.

Lee, H.L. (2004), "The triple-a supply chain", *Harvard Business Review*, October, pp. 102-12.

Matthyssens, P. and Vandenbempt, K. (1998), "Creating competitive advantage in industrial services", *Journal of Business & Industrial Marketing*, Vol. 13 Nos 4/5, pp. 339-55.

Object Management Group (2009), "Unified modelling language", available at: www.omg.org

Ojasalo, K. (2007), "Development industrial services-an empirical study", *The Business Review*, Vol. 7 No. 1, pp. 58-62.

Pinto, S. and Johnston, R. (2009), "Understanding specification for the control of service", paper presented at 16th International Annual EurOMA Conference, Göteborg, Sweden.

Robertson, S. and Robertson, J. (2006), *Mastering the Requirements Process*, Addison-Wesley, Reading, MA.

Roth, A.V. and Menor, L.J. (2003), "Insights into service operations management: a research agenda", *Production and Operations Management*, Vol. 12 No. 2, pp. 145-64.

Smeltzer, L.R. and Ogden, J.A. (2002), "Purchasing professionals' perceived differences between purchasing materials and purchasing services", *Journal of Supply Chain Management*, Vol. 38 No. 1, pp. 54-70.

Sommerville, I. and Sawyer, P. (1997), *Requirements Engineering: A Good Practice Guide*, Wiley, New York, NY.

Standish Group (1995), *The Chaos Report*, Standish Group, Boston, MA.

van der Valk, W. and Rozemeijer, F. (2009), "Buying business services: towards a structured service purchasing process", *Journal of Service Marketing*, Vol. 23 No. 1, pp. 3-10.

Wynstra, F., Axelsson, B. and van der Valk, W. (2006), "An application-based classification to understand buyer-supplier interaction in business services", *International Journal of Service Industry Management*, Vol. 17 No. 5, pp. 474-96.

Zomerdijk, L. and Voss, C.A. (2011), "NSD processes and practices in experiential services", *Journal of Product Innovation Management*, Vol. 28, pp. 63-80.

## Further reading

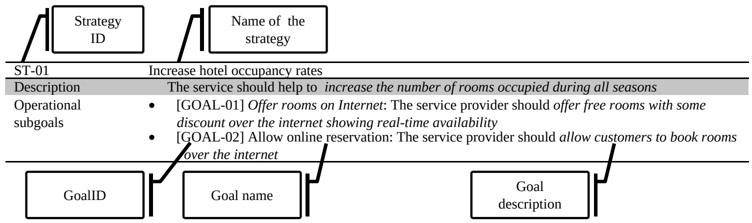
van Mossel, H.J. and van der Valk, W. (2009), "Securing customer satisfaction through component service specifications: purchasing maintenance services for social rented housing", *Journal of Purchasing and Supply Management*, Vol. 14 No. 4, pp. 241-52.

## Appendix. Examples from the proposed requirements document

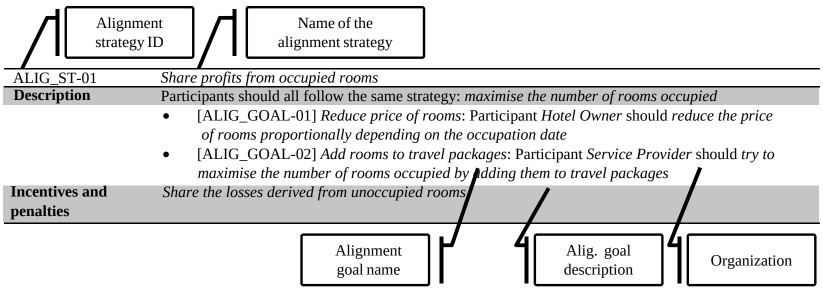
In this appendix, a number of examples, referred to in the text earlier, are given of elements of a hypothetical requirements document based on a hotel chain seeking to outsource its sales online sales service.

### *1. Strategy, goals and alignment templates*

The customer's goals and the strategy linking them with the (service-related) operational processes that must be executed for them to be achieved (Section 4.1 of the requirements document) must be specified. The following strategy and goals templates can be used for this:

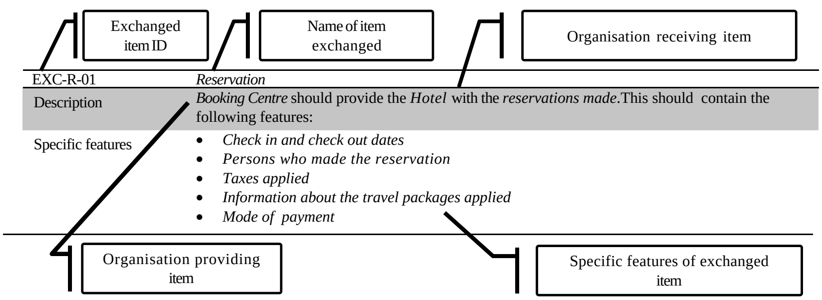



The following template is proposed for documenting alignment requirements (Section 4.2 of the requirements document):



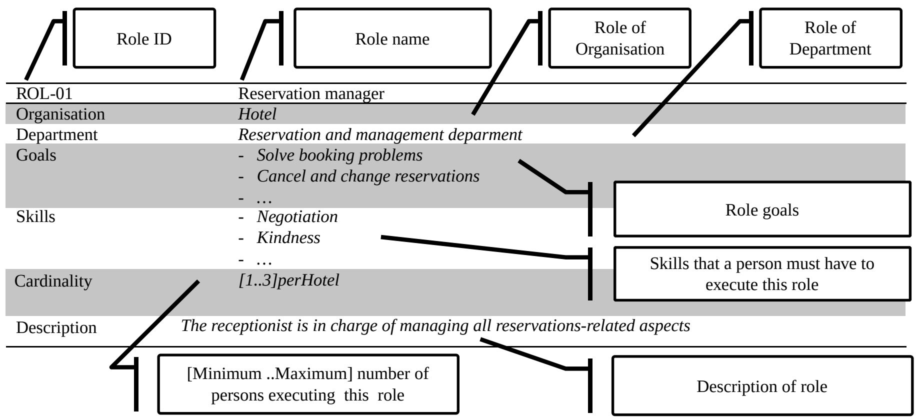

*2. Exchanged items requirements templates*

This section addresses Section 5.1 of the requirements document. The template for items exchanged between roles in the services must describe all the items exchanged. These items might be information or goods. Note that exchanged intangible items must be represented by means of quality of the service (QoS) requirements, e.g. high-customer satisfaction level in 90 per cent of the cases, or by means of interaction cases, e.g. ability to transfer money from one account to another.




*3. Participating role templates*

This section shows the proposed template for the roles involved in the BS transaction (Section 5.2.1 of the requirements document):




## 4. Interaction diagrams

This section explains how to cover Section 5.2.2.1 of the requirements document.

Interactions diagrams are used to illustrate which actors must interact and which use cases drive their interactions. These diagrams are used as a graphical summary or index that show all the actors and interactions between them in just one place. This is needed because the number of interactions cases (textual templates spread in several pages) is usually high and is difficult to get an overall idea of the whole system. An example of this diagram is shown in Figure A1; as can be seen, actor icons are used to represent roles and ellipses are used to represent the interaction cases.

In the SE field, these diagrams are called use case diagrams[1] and are part of UML (software modelling language) (Object Management Group, 2009).

## 5. Interaction case templates

These templates show the interactions between buyers and providers (Section 5.2.2.2 of the requirements document), basing on software use case templates (Cockburn, 1997, 2000). (Use cases represent the interactions between users and software, interaction cases represent the interaction between two business actors.) A sequence of actions (steps) representing the standard procedure for the interaction must be provided to specify a use case. Only the standard procedure should be shown because the software field has shown that complex sequences of steps where several different paths can be taken to complete the interaction can result in unclear specifications. This item has been renamed as interaction cases, since they represent the interactions between different roles in BS.

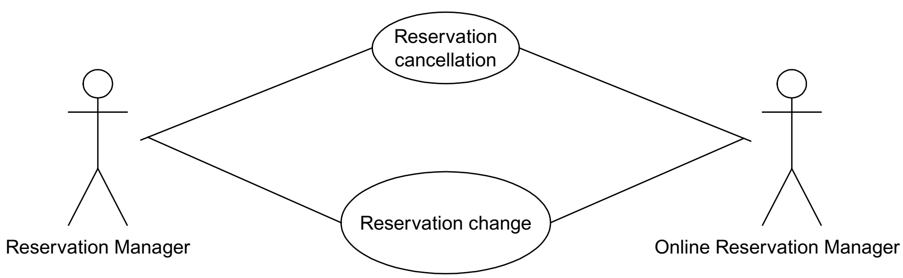
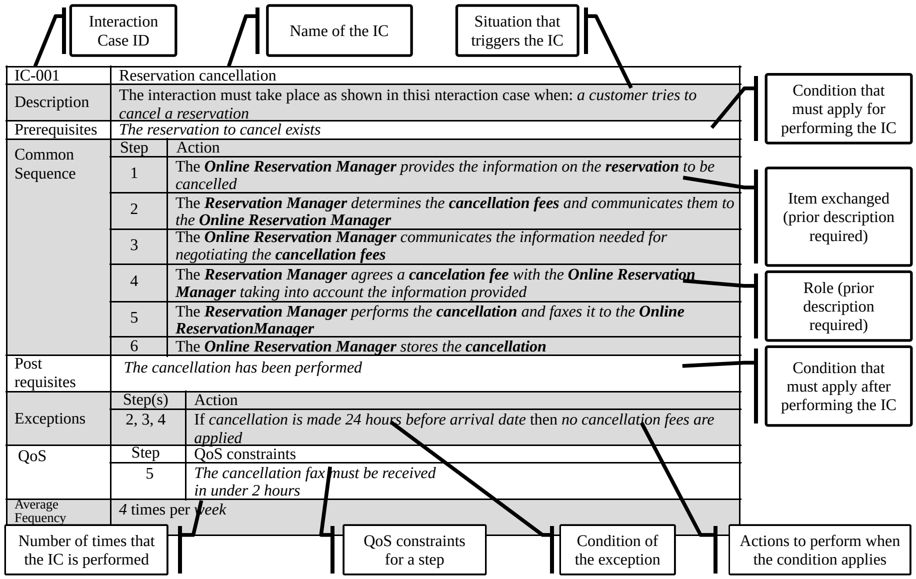



**Figure A1.**
Interaction diagram using UML use case diagrams

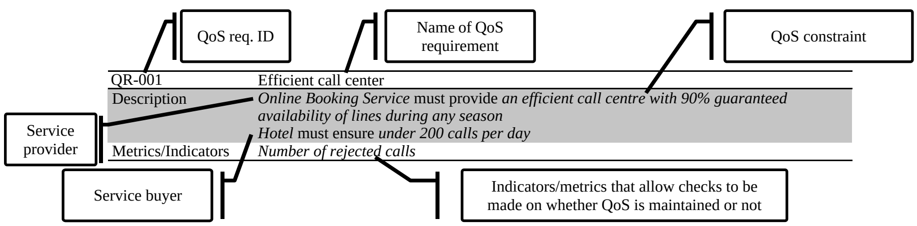



| IC-001 | Reservation cancellation | |
|---|---|---|
| Description | The interaction must take place as shown in thisi nteraction case when: *a customer tries to cancel a reservation* | |
| Prerequisites | *The reservation to cancel exists* | |
| Common Sequence | Step | Action |
| | 1 | The ***Online Reservation Manager*** *provides the information on the* ***reservation*** *to be cancelled* |
| | 2 | The ***Reservation Manager*** *determines the* ***cancellation fees*** *and communicates them to the* ***Online Reservation Manager*** |
| | 3 | The ***Online Reservation Manager*** *communicates the information needed for negotiating the* ***cancellation fees*** |
| | 4 | The ***Reservation Manager*** *agrees a* ***cancelation fee*** *with the* ***Online Reservation Manager*** *taking into account the information provided* |
| | 5 | The ***Reservation Manager*** *performs the* ***cancellation*** *and faxes it to the* ***Online ReservationManager*** |
| | 6 | The ***Online Reservation Manager*** *stores the* ***cancellation*** |
| Post requisites | *The cancellation has been performed* | |
| Exceptions | Step(s) | Action |
| | 2, 3, 4 | If *cancellation is made 24 hours before arrival date* then *no cancellation fees are applied* |
| QoS | Step | QoS constraints |
| | 5 | *The cancellation fax must be received in under 2 hours* |
| Average Fequency | *4 times per week* | |

*6. Quality of service requirements templates*

This section explains how to cover section.

The following template can be used to specify the QoS (Section 5.3.1 of the requirements document) (notice that legal, cultural, moral and ethical aspects must be documented in a similar way):




| QR-001 | Efficient call center |
|---|---|
| Description | *Online Booking Service* must provide *an efficient call centre with 90% guaranteed availability of lines during any season*<br>*Hotel* must ensure *under 200 calls per day* |
| Metrics/Indicators | *Number of rejected calls* |

Note that the interaction-related part of the QoS will have been documented using interaction cases.

*7. Traceability matrixes*

Traceability matrixes show dependencies between items in the specification. Conventional matrixes show which exchanged item requirements and functional and non-functional requirements are needed for each of the service's goals to be achieved (Section 4 of the requirement document). The goals, identified by their IDs, are set out in rows in a table and the IDs of the requirements in the requirements catalogue (Section 5 of the requirements document) in columns. When the goal relates to a requirement, a cross is placed in the corresponding cell. The template is as follows:

| | Req-01 | Req-02 | … |
|---|---|---|---|
| Goal-01 | X | | … |
| Goal-02 | | X | … |
| … | … | … | … |

Note that goals are achieved through interactions between participants in the service. Therefore, checks must be made to verify whether all goals are detailed by means of interaction cases and all relating items, namely exchanged items requirements, roles, QoS constraints, etc.

*8. Information required in all templates for management specification changes*
Each item must have an ID so that dependencies with other items can be established:

| ID | e.g. IC-001 |
|---|---|
| Dependencies | List of IDs of other templates |

In order to manage changes in requirements and to decide which should be used for the next version of the service the following information is needed:

| | |
|---|---|
| Importance | [critical, major, normal, minor, insignificant] |
| Urgency | [very urgent, urgent, needed, would be good] |
| Stability | [very stable, stable, unstable, very unstable] |

To identify those involved in the specification of each item and whether the requirement has been validated by the buyers, contains errors, is a draft, has been implemented, etc. all the templates must contain the following information:

| Version | Version number (date) |
|---|---|
| Authors | Authors' names |
| Sources | Names of people providing the information |
| Source documentation | Documents used for specifying the requirement |
| State | [Validated by user, pending, draft, implemented, etc.] |

Finally, all items can also have a comments field:

| | |
|---|---|
| Comments | … |

## Note

1. Use case diagrams" in the SE field relates a software systems with their users (actors). However, in the BS, as two or more actors are related by means of an interaction, we propose to call them "interaction diagrams".

## About the authors

Dr Joaquin Peña-Siles has been a Lecturer of Software Engineering at the University of Seville since 2000. He received his Doctorate in Software Engineering from the University of Seville in 2005 and also holds an Executive Masters in Business Administration (2010) and is the CEO and Founder of a spin-off firm at the University of Seville. This firm is devoted to the application of research findings. He is the author/coauthor of more than 60 articles in prestigious journals and

conference proceedings in the field of operations management, business process management, service science, Enterprise 2.0 and software engineering. He collaborates with reserchers in US and other European universities and research centres, such as NASA Software Engineering Laboratory, where he holds several patents. He has participated in many research projects and holds several awards for his teaching and research. Other recognitions include: PhD with Honors, best project of the Entrepreneurial Project Awards of 2007 (University of Seville), mentor of several student teams at national competitions (Deloitte Business Game, Microsoft Imagine Cup, etc.), and NASA Awards for Technology Transfer 2006. Dr Joaquin Peña-Siles is the corresponding author and can be contacted at: joaquinp@us.es

Maria del Mar González Zamora, Dr of Business Management, graduated in Economics and Business Science at the University of Seville in 1993. Presently she is an Associate Professor at the University of Seville, where she has been teaching for the last 17 years in the areas of Operations Management and Business Simulation with System Dynamics. Her research has attracted funding from both within the European Union framework and at the national level. She has coauthored articles published in journals, such as *Journal of Operations Management*, *Journal of Service Management*, as well as in national and international conferences (European Management Association, Production and Operations Management, System Dynamics, etc.). Her current research focuses on service operations management and the design and creation of business games for transparent learning environments based on system dynamics. She is a member of various scientific associations: EurOMA (European Operations Management Association) and ACEDE (Economics and Business Management Scientific Association). She is the recipient of the following awards: the 2001 Wickham Skinner Award on Teaching Innovation Achievements, awarded by the Production and Operations Management Society; nominated for the prestigious European Ist Prize (2001); 2002 AEDEM-MED Academy of Management Award on Management Education; Honourable Mention at the 2002 Instructional Award Competition (Decision Science Institute).

Dr José A.D. Machuca, Dr of Industrial Engineering, Master in Business Administration and Master in Industrial Engineering, is Professor of Operations Management at the University of Seville, Spain. He is Director of GIDEAO Research Group, Former Head of Department (1982-1992) and Former Vice Dean (1983-1989), Vicepresident EurAfrica of POMS (Productions and Operations Management Society) and member of the EurOMA Board (European Operations Management Association). Coordinator of the European Thematic Network for Excellence in Operations and Supply Chain Management Education, Research and Practice (Thenexom). Dr Machuca is the Director of three European projects, seven national projects and two Projects of Excellence (Andalusian Research Plan). He is on the Editorial Advisory Board of *Journal of Operations Management*, and *International Journal of Operations & Production Management*. He is also Regional Editor for *Operations Management Research* and on the Editorial Review Board of *Production and Operations Management* and *International Journal Manufacturing Technology Management*. Dr Machuca is author/coauthor of eight books and editor/coeditor of eight journal special issues, more than 50 articles and 60 book chapters. He has published in journals such as: *JOM*, *POM*, *IJOPM*, *IJPE*, *JPR*, *HBR*, *Transportation Research* and *SDR*. His awards include: 2001 Wikham Skinner Award honouring Teaching innovation achievements (POMS), Nominated for the 2001 European Ist Prize, 2002 MED-Academy of Management-AEDEM Award in Management Education, Honourable Mention 2002 Instructional Innovation Award (DSI), 2003 POMS and Indiana CIBER Best Case International Award, 2005 Business Week/ECCH European Best Case in Operations Management Award, Honourary Doctorate by the Universidad Privada del Norte (Perú); 2009 Andalucia Research Award-Ibn Al Jatib, 2009 University of Sevilla FAMA Research Award; Outstanding Professor, University of Seville (2003-2004, 2004-2005).