Grant Agreement N° 215483

Title:	*Initial Definition of User Patterns*
Authors:	*UniDue, Tilburg, City, FBK, Lero, POLIMI, UoC, UPM, USTUTT, VUA*
Editor:	*Qing Gu (VUA), Patricia Lago (VUA), Michael Parkin (Tilburg)*
Reviewers:	*Mañuel Carro (UPM), Pierluigi Plebani (POLIMI), Raman Kazhamiakin (FBK)*
Identifier:	*PO-IA-3.1.4*
Type:	*Project Outcome (PO)*
Version:	*1.4*
Date:	*June 15, 2010*
Status:	*Final*
Class:	*External*

Management summary

S-Cube's WP-IA-3.1 ("Integration Research Framework: Baseline & Definition") has the objective of defining a coherent, holistic framework for service engineering and adaptation - the Integrated Research Framework, or IRF. The framework developed as part of this workpackage should cater for various types of service-based application (SBA) stakeholders, or parties with an interest or concern in the design, development and operation of SBAs. This interim deliverable identifies an initial set of stakeholder types which will be refined and expanded as part of Task T-IA-3.1.3 later in the project.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n 215483 (S-Cube).

Members of the S-Cube consortium:

University of Duisburg-Essen (Coordinator) – UniDue	Germany
Tilburg University – Tilburg	Netherlands
City University London – CITY	U.K.
Consiglio Nazionale delle Ricerche – CNR	Italy
Center for Scientific and Technological Research – FBK	Italy
French Natl Institute for Research in Computer Science and Control – INRIA	France
The Irish Software Engineering Research Centre – Lero	Ireland
Politecnico di Milano – Polimi	Italy
MTA SZTAKI – Computer and Automation Research Institute – SZTAKI	Hungary
Vienna University of Technology – TUW	Austria
Universit Claude Bernard Lyon – UCBL	France
University of Crete – UOC	Greece
Technical University of Madrid – UPM	Spain
University of Stuttgart – USTUTT	Germany
VU University Amsterdam – VUA	Netherlands
University of Hamburg – UniHH	Germany

Published S-Cube documents

These documents are all available from the S-Cube Web Portal at `http://www.s-cube-network.eu/`

Foreword

The objective of WP-IA-3.1 is to define a coherent, holistic framework for service engineering and adaptation, or the IRF. The framework will integrate principles, techniques and methodologies for the engineering and adaptation produced by S-Cube workpackages WP-JRA-1.1 (engineering principles, techniques and methodologies for hybrid, service-based applications), WP-JRA-1.2 (adaptation and monitoring principles techniques and methodologies for service-based applications) and WP-JRA-1.3 (end-to-end quality provision and SLA conformance), and will encompass the technologies of the horizontal technology layers for service-based applications (i.e., service infrastructures, service composition and co-ordination and business process management covered in S-Cube's Joint Research Activities WP-JRA-2.1, WP-JRA-2.2 and WP-JRA-2.3).

The framework developed as part of this workpackage — the Integrated Research Framework, or IRF — should cater for various types of service-based application (SBA) stakeholders, or parties with an interest or concern in the design, development and operation of SBAs. For example, an SBA developer may act as a service composer, service manager or service consumer, or a combination of all three. These different stakeholder types will exploit different categories of knowledge of the technical aspects of service-based applications and will adopt different engineering and adaptation methodologies and processes.

It is the goal of this interim deliverable to take into account the co-existence of the various types of stakeholders involved in the engineering, adaptation and consumption of service-based applications and identify an initial set of stakeholder types which will be refined and expanded later in the project (specifically as part of Task T-IA-3.1.3 "Analysis of User Patterns & Methodologies" and in CD-IA-3.1.6 "Final Definition of User Patterns & Methodologies", due in Month 39 of the project).

The editors of this document would like to take this opportunity thank Dinh Khoa Nguyen (Tilburg) for his efforts and contributions to the S-Cube project during the preparation of this deliverable.

Contents

Chapter 1

Introduction

The engineering, adaptation and consumption of services involves many different parties, from the designers of the service-based application to the providers of the service components and eventual end-users of a service-based application. In this document we define the collective terms for these parties to be *stakeholders*, as they have an interest or concern in these processes. The term stakeholder is used as it is a more generic term than *user*, which has different connotations and may be a type of stakeholder. We have chosen to focus on stakeholders involved in the design, implementation and ongoing operation of service-based systems since the Integrated Research Framework (IRF) developed in this work package should cater for various types of service-based application (SBA) stakeholders, or parties with an interest or concern in the design, development and operation of SBAs.

Therefore, this deliverable seeks to provide an initial taxonomy of the different roles stakeholders can take in the design and execution of service-based systems to ensure the research output of S-Cube meets the needs and requirements of these stakeholders. In the description of work (DoW) [2], this deliverable is described as also providing a summary of the methodologies tailored for each of the stakeholders identified. However, in this deliverable we have chosen to concentrate solely on the definition of stakeholders in this document and the methodologies will be presented in the follow-up deliverable to this interim deliverable, CD-IA-3.1.6 "Final Definition of Users Patterns & Methodologies".

1.1 Context

The previous work of this S-Cube activity has, in deliverable CD-IA-3.1.1 [5], provided a baseline version of the integrated engineering and adaptation framework (the IRF) from which the interfaces between the technology layers and principles, techniques and methodologies were derived and documented in [4]. The first version of the IRF, which took the research results from the S-Cube's JRA-1 and JRA-2 research activities and mapped them onto those research interfaces, was produced in M21 and is described in [6]. As described above, this deliverable is an initial definition of service-based application stakeholders and, as we will describe in the conclusions, will be used to focus further efforts in this area to meet the gaps found in the analysis of this initial set of definitions and to provide a focal point for the definition of the methodologies that, as described, are to follow in the final definition of stakeholder patterns and methodologies provided in S-Cube deliverable CD-IA-3.1.6 "Final Definition of User Patterns & Methodologies", due in Month 39 of the project (June 2011).

1.2 Purpose/Objective of the Deliverable

The purpose of the Integrated Research Framework (IRF) developed in WP-IA-3.1 is to cater for the needs of various types of stakeholders. As we will show in this document, there are different types of stakeholders involved in different phases of the SBA life-cycle, each of which has different roles, goals, and competences when developing or interacting with service-based applications. These different types of stakeholder will exploit different categories of the knowledge, results and methodologies integrated and presented within the IRF. In particular, such stakeholder types may include:

- Consumers and users of services-based applications including experienced and inexperienced end-users and citizens;
- Service composers and users involved in the system design, such as software engineers, system integrators and architects, business experts.

This document will reflect the initial findings and results of the IA-3.1 workpackage's task T-IA-3.1.3 "Analysis of the User Patterns and Methodologies". This task aims to reveal these different stakeholder types and to associate the appropriate design and management methodologies corresponding to the activities and roles those stakeholder types perform. The activity aims also to identify different types of the human involvement in the SBA life-cycle, ranging from end-users, to responsible of IT infrastructures, engineers, etc.

The results of this activity, and specifically the results of the presented deliverable, will be obtained in close collaboration with the other S-Cube work packages and activities. For example:

- Together with the **Integrated Knowledge Model** (IA-1.1), this task will define the taxonomy of the different stakeholders involved in the SBA engineering and execution, as well as the corresponding methodologies.
- The identification of the stakeholder types is based on the investigation and research activities within the research work packages, and in particular WP-JRA-1.1 **Service Engineering and Design**, where the reference life-cycle of the adaptable SBAs is defined, studied, and refined. Indeed, the different types of stakeholder will be involved in different phases of the SBA life-cycle, having different roles and exploiting different methodologies.
- Different stakeholders perform different activities and will be provided with the different methodologies and roles that will be identified and studied within the other research WPs, where the corresponding aspects will be addressed. That is, the **Service Monitoring and Adaptation** workpackage (WP-JRA-1.2) will contribute to the stakeholder types from the perspective of the monitoring and adaptation activities and the involvement of the humans in these activities; **End-to-end Quality Provisioning** (WP-JRA-1.3) will concentrate on the activities related to the service quality modeling and measuring during the SBA life-cycle, etc.

The stakeholder types obtained will also contribute to the **Integrated Research Framework** as these roles and patterns can be mapped onto the elements of the IRF, used to define service-based engineering knowledge and methodologies and contribute to the objectives of the IRF and to the research challenges of the S-Cube projects. Specifically, the following S-Cube research challenges (see the S-Cube Vision Document) are relevant for task T-IA-3.1.3 and this deliverable:

- *Definition of a coherent life cycle for adaptable and evolvable SBA*. As this challenge aims the definition and refinement of the SBA life-cycle, it will also address the different human activities in its different phases, as well as the different types of such an involvement and the corresponding design activities.
- *Exploiting stakeholder and task models for automatic quality contract establishment*. This challenge explicitly addresses the ways of involvement of certain types of stakeholder in the process of SBA provisioning and service integration.
- *Context- and HCI-aware SBA monitoring and adaptation / HCI and context aspects in the development of service-based applications*. The research activities devoted to this challenge will study the usage of the human-related information and activities in the process of design, monitoring, and adaptation of SBA, providing novel methodologies related to the end-user management.

- *Mixed initiative SBA adaptation*. The challenge aims to focus different forms of the human involvement into the SBA adaptation process, ranging from human-in-the-loop adaptation to completely autonomous adaptation, and to provide the appropriate adaptation principles and methodologies corresponding to these forms.

Finally, the defined methodologies will be also evaluated and validated based on the case studies and scenarios defined in the **IRF Validation** work package (IA-3.2).

1.3 Structure of this deliverable

The remainder of this deliverable is structured as follows: in Chapter 2 we present the service and service-based application lifecycle model S-Cube has developed to address both the classical software development process and the new requirements arising from the necessary adaptation of services and service-based applications. In this deliverable, the S-Cube lifecycle will be used to demonstrate the co-existence of various types of stakeholder involved in the engineering, operation, adaptation and consumption of service-based applications.

Chapter 3 describes the methodology used to collect this initial set of stakeholder types and the initial results it produced. A brief analysis of the results is presented in Chapter 4, which concentrates on demonstrating the coverage of the S-Cube lifecycle by the consolidated stakeholder types in Section 4.1. Chapter 5 provides a conclusion with Section 5.1 giving suggestions for future work.

Chapter 2

S-Cube Research Framework & Service-Based Application Lifecycle

The S-Cube research framework places an emphasis on avoiding the pitfalls of deploying services in an uncontrolled manner and provides a holistic and solid approach for service development in an orderly fashion so that services can be efficiently combined into service-based systems. The S-Cube research framework views service-based systems as an orchestrated set of service interactions and adopts a broader view of its impact on how the service-based solutions are designed, what it means to assemble them from disparate services, and how deployed services-oriented systems can evolve and be managed. This requires addressing common concerns such as the identification, specification and realization of services, their flows and composition, as well as ensuring the required quality levels.

2.1 Motivation

In their early use of SOA, many enterprises assumed that they could port existing components to act as Web services by merely creating wrappers and leaving the underlying components untouched. Since component methodologies focus on the interface, many developers assume that these methodologies apply equally well to service-oriented environments. As a consequence, implementing a thin SOAP/WSDL/UDDI layer on top of existing systems or components that realize the services is by now widely practiced by the software industry. Yet, this is in no way sufficient to construct commercial strength service-based systems. Unless the nature of the component makes it suitable for use as a service, and most components are not suited to this, for instance, because they are tightly coupled to other components, it takes serious thought and redesign effort to properly deliver a components functionality through a service. While relatively simple Web services may be effectively built that way, a service-based development methodology is required to specify, construct, refine and customize highly flexible service-based systems from internally and externally available components and services. More importantly, older software development paradigms for object-oriented and component-based development cannot be blindly applied to SOA and services. This is the motivation for the S-Cube service lifecycle; the lifecycle model builds on established practices from software engineering to provide a general structure within which many different service-specific principles and techniques to be positioned in order to facilitate the next generation of service-engineering methodologies.

2.2 S-Cube Lifecycle View

The service lifecycle model envisioned by the S-Cube framework, shown in Figure 2.1, captures an iterative and continuous method for developing, implementing, and maintaining services in which feedback is continuously cycled to and from phases in iterative steps of refinement and adaptations of all three

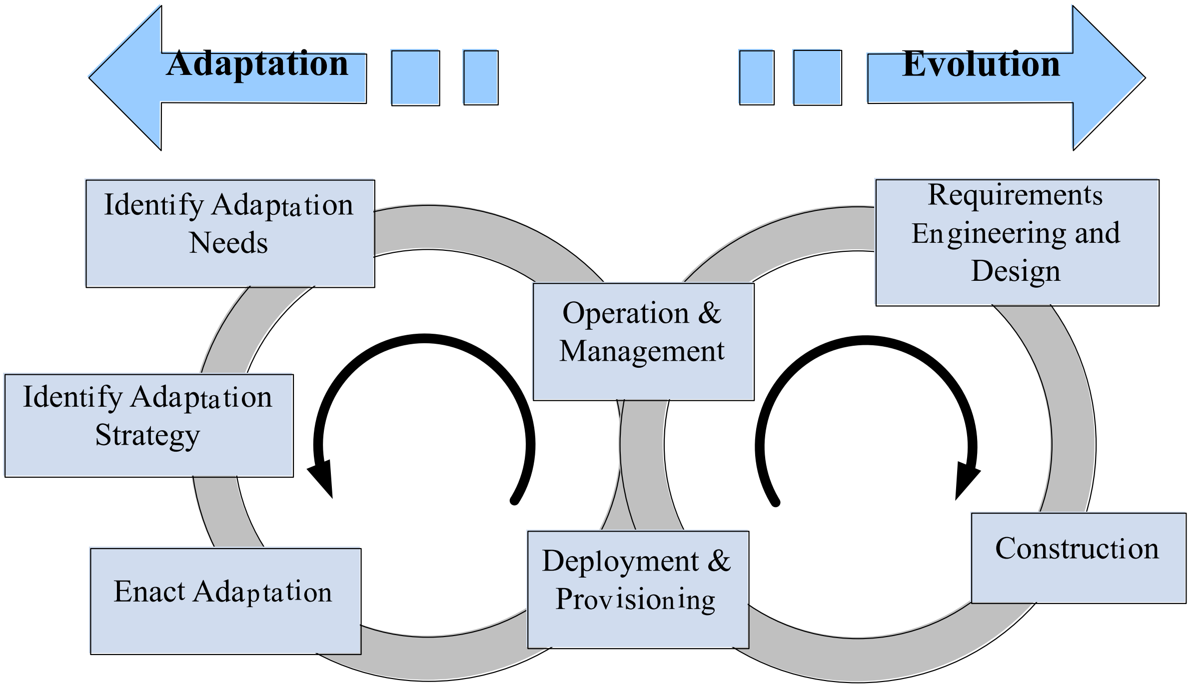

Figure 2.1: The S-Cube Lifecycle View

layers of the technology stack. This lifecycle facilitates designing solutions as assemblies of services in which the assembly description is a managed, first-class aspect of the solution, and hence, amenable to analysis, change, and evolution. The method accommodates continuous modifications of service-based systems and its quality (e.g., QoS and KPIs) at all layers. Continuous modifications (evolutions and adaptations) are based on monitoring and measurement of service execution against SLAs and quality goals.

As shown in Figure 2.1, the development cycle (the right-hand cycle) addresses the classical development and deployment lifecycle phases, including requirements and design, construction and operations and management. The second cycle (the left-hand cycle) extends the classical lifecycle by explicitly defining phases for addressing changes and adaptations. As a result, the S-Cube service lifecycle allows the: continuous detection of new problems, changes and needs for adaptation; identify possible adaptation strategies, and; enact adaptation strategies. Once service-based systems (or parts thereof) have been adapted, they will be re-deployed and re-provisioned and put into operation. More details on the S-Cube lifecycle can be found in S-Cube deliverable CD-JRA-1.1.2 “Separate design knowledge models for software engineering and service-based computing” [3].

Chapter 3

Research Method & Initial Stakeholder Types

This section provides a characterization of the different types of stakeholders that are involved with the design and execution of service-based applications that were discovered through a methodology that was designed to collect, collate and compare roles identified by researchers from the S-Cube network from their own research and that of the wider community. The methodology is described in Section 3.1, with the interim results of the process described in Section 3.2.

3.1 Stakeholder Definition Methodology

To find the initial set of stakeholders involved in the design, execution and consumption of service-based applications and systems we used the methodology described below. This process was designed to collect, collate and compile a large set of specifications gathered from S-Cube partners and to reduce the large set of varied responses to a manageable collection.

1. The deliverable leader solicited stakeholder types from workpackage participants using the stakeholder specification template shown in Table 3.1 (page 11).
2. The template was completed by each workpackage participant for each stakeholder type they have identified through their own research or from the relevant literature and returned to the deliverable editors. The full set of 'raw' results are presented in Appendix A.
3. To create an initial grouping and ordering of the stakeholder types, we used the information in each specification returned to classify each stakeholder type as belonging to one or more of the following broad categories, which were taken from [1]:

    - The *Service Providers* category are parties that produce and publish services which are ready to be executed and who are owners services. They are responsible for implementing services as well as maintaining services.
    - The *Service Composers* category compose existing services or software applications for achieving certain business goals and provide the composite services for internal or external usage.
    - The *Application Builders* category integrates services into an application which eventually fulfill the requirements of the end users.
    - The *Application Clients* category refer to the end-users who use the application to achieve certain goals.
    - *Supporting Roles* are stakeholder types that are not directly involved with the service lifecycle (e.g., Project Managers).

The classified types were sorted alphabetically by stakeholder type within their classifications. The results can be seen in the first three columns of Table 3.2 (page 12).

4. To reduce the classified types to a more manageable group we gathered together types with a similar function. I.e.: the stakeholder type *Business Process Architect* has the context *Business Process* and function *Architect*. In this step we separated the functions from the context of each stakeholder and grouped similar functions. When we carried out this exercise, we found many similar functions, therefore the functions of these similar stakeholder types were 'aligned' by replacing similar functions with synonyms, or a word or phrase with a similar meaning to that given. In the methodology we used the following synonyms to align stakeholder with similar functions:

   - *Analyst* was treated as a synonym for *Modeler*.
   - *Designer* was considered a synonym for *Architect*.
   - *Enterprise* was judged a synonym for *Business*.
   - *Service-Based Application (SBA)* was used as a synonym for *System*, *End-point Service-Based System* and *Service Network*.

5. Stakeholder types with the same, harmonized function were merged to provide the initial set of stakeholder types (column four of Table 3.2).

3.2 Results

The results of applying the methodology described in Section 3.1 (above) to the results shown in Appendix A can be seen in Table 3.2. As can be seen, the methodology has reduced the 50 initial stakeholder types to 19 consolidated types. The next step is to use these consolidated stakeholder types to determine their relationship to and coverage of the S-Cube lifecycle described in Chapter 2.

Service Engineering Stakeholder Specification	
Contributor	*Who filled in the table*
Type of Stakeholder	*e.g., service developer, business analyst*
S-Cube Lifecycle Phases	*In which phases of the S-Cube lifecycle (see Chapter 2) is this stakeholder involved?*
Activities	*In which activities (within the phases above) is this stakeholder involved in?*
Description	*Description about the responsibility of the stakeholder within the activities above*
Peculiarity to Service Engineering	*What are the characteristics of stakeholder participation in service engineering, e.g., in comparison with the participation in conventional software engineering approaches? In other words, what does this stakeholder do differently than traditional software engineering?*
Interactions with other Stakeholders	*Any interactions exist? For what purposes?*
References	*e.g., in any service engineering methodologies, S-Cube deliverables, case studies, publications... (full citation)*
Glossary	*Terms from S-Cube Knowledge Model*
Related Research Challenges	*Any gap between this stakeholder specification and the support from existing service engineering methodologies could be identified? How to fill the gap?*

Table 3.1: Service Engineering Stakeholder Template Used to Collect Identified Users of Service-Based Applications

Stakeholder Type	Submitted By	Category	Consolidated Stakeholder Type
Application Architect	City	Application Builders	SBA Architect
Application Designer	POLIMI		
Business Process Engineer	City		
SBA Assembler	City		
SBA Developer	Lero		
SBA System Builder	City		
System Analyst	City		SBA Modeler
Service Network Modeler	USTUTT		
Business Process Architect	Tilburg		
Application Builder	City		Application Developer
Application Developer	POLIMI		
Business Analyst	City		Business Analyst
Business Analyst	Lero		
Business Process Analyst	USTUTT		Business Process Analyst
Business Process Modeler	Tilburg		
Business Process Manager	Tilburg		Business Process Administrator
Business Process Owner	USTUTT		
Domain Expert	City		Domain Experts
Domain Experts	UniDue		
Experts for User Interfaces	UniDue		
Requirements Engineer	City		SBA Engineer
Service Engineer	Lero		
End User	FBK	Application Clients	End User
End User	City		
Direct User	UniDue		Service Consumer
Indirect User	UniDue		
Service Consumer	POLIMI		
Service Consumer	UoC		
Composition Designer	UoC	Service Composers	Composition Designer
Service Composition Designer	UPM		
Negotiation Agent	UoC		Negotiation Agent
Enterprise Architect	City	Service Providers	Service Architect
Service Architect	USTUTT		
SOA Domain Architect	UPM		
SOA Platform Architect	UPM		
Service Deployer	USTUTT		Service Deployer
Service Designer	City		Service Designer
Service Designer	POLIMI		
Service Designer	UPM		
Service Developer (1)	City		Service Developer
Service Developer (2)	City		
Service Developer	POLIMI		
Service Developer	USTUTT		
Service Developer/Provider	Lero		
Service Provider	POLIMI		Service Provider
Service Provider	UoC		
Change Manager	UniDue	Supporting Roles	Manager
Project Manager	City		
Technology Consultants & Suppliers	UniDue		Supporting Expert
Lawyers & Data Privacy Officers	UniDue		

Table 3.2: Results from Research Methodology: Classified, Sorted and Consolidated Stakeholders

Chapter 4

Analysis

Having found the initial set of consolidated stakeholder types in Chapter 3, this section will analyze the results by considering their relationship to and coverage of the S-Cube lifecycle shown in Figure 2.1 (page 8).

4.1 Lifecycle Coverage

To provide the information of where the initial set of consolidated stakeholder types mapped to the S-Cube service lifecycle, we inspected the lifecycle phases each of the submitted stakeholder types and, for each stakeholder type, recorded which lifecycle phases they applied to. Note that where stakeholder types had been consolidated the lifecycle phases recorded for the consolidated type are the merged set of lifecycle phases of each of the submitted stakeholder types — i.e., if two S-Cube partners have defined the same stakeholder type but asserted they belong in different phases of the S-Cube lifecycle, when the stakeholder was consolidated (Step 5 of the methodology described in Section 3.1)

The results of mapping the consolidated stakeholder types to the phases of the S-Cube lifecycle are shown in Table 4.1.0.2. As can be seen in the table, each of the phases of the lifecycle is covered, with some more comprehensively than others. The last row of the table shows the coverage of each lifecycle phase as a percentage relative to the total number of consolidated stakeholder types; reviewing the figures we can see the lifecycle phases of 'early requirements engineering', 'requirements engineering and design' and 'identification of adaptation needs' all have good coverage with respect to stakeholder types. The engineering lifecycle phase of 'construction' and 'enact adaptation' have the least amount of coverage (less than 50%) and it is these stages of the lifecycle we recommend more investigation into with respect to identifying the stakeholders involved in these phases in Section 5.1.

4.1.0.1 Mapping Stakeholders to the S-Cube Lifecycle

To explore the results further and visually demonstrate the coverage, we can take the results from Table 4.1.0.2 and map each of the consolidated stakeholder types onto the phase in the S-Cube lifecycle where they participate. Such a mapping is shown in Figure 4.1.

Looking at Figure 4.1, what is of interest is how certain stakeholder types feature more prominently than others: for example, the *SBA Architect* and *Manager* types are omnipresent throughout both lifecycles, whilst *Supporting Experts* are only present in the design and engineering and operations and management phases. Further, the general *Application Builder* stakeholder types are involved more in the engineering and design phases than other phases, whilst the *Service Provider* stakeholders are more prominent in the operations and management and deployment and provisioning phases.

These initial results confirm much of what we would expect regarding specific stakeholder types and their involvement in the service and service-based application lifecycle phases; the requirements engineering and design phases require the input and experience of many stakeholder types to ensure the service or service-based application being developed or built to ensure the service is useful to its intended end-users. Conversely, operating and managing a service or SBA or enacting a previously decided adaptation strategy requires relatively fewer roles to complete, which is what we would expect from intuition.

4.1.0.2 Summary

When considering the differences between the stakeholders involved in left-hand and right-hand side of Figure 4.1 (i.e., at the differences between the stakeholder present in adaptation and evolution phases of the S-Cube lifecycle), we can see that in the adaptation side of the S-Cube lifecycle a much broader range of roles are involved than in the evolution phases; for example, it appears groups such as application clients and service composers are more present in the adaptation phase than in the evolution phase. Conversely, we find that groups such as application builders are represented to a greater extent in the evolution phases rather than the adaptation phases. What we think this demonstrates is how the evolution phase is focussed much more on the engineering and design and implementation processes than the adaptation lifecycle which requires the input of the service-based application's end-users, such as application clients and service consumers.

		Evolution Lifecycle					Adaptation Lifecycle		
Category	Consolidated Stakeholder Type	Early Re-quirements Engineering	Requirements Engineering & Design	Construction	Deployment & Provision	Operation & Management	Identify Adaptation Needs	Identify Adaptation Strategies	Enact Adaptation
Application Builders	SBA Architect	✓	✓	✓	✓	✓	✓	✓	✓
	SBA Modeler	✓	✓		✓		✓	✓	
	Application Developer	✓	✓	✓	✓				
	Business Analyst	✓	✓				✓	✓	
	Business Process Analyst	✓	✓	✓			✓	✓	
	Business Process Administrator	✓	✓		✓		✓	✓	
	Domain Expert	✓	✓						
	SBA Engineer	✓	✓			✓			
Application Clients	End User	✓				✓	✓	✓	✓
	Service Consumer	✓	✓		✓	✓	✓		✓
Service Composers	Composition Designer	✓		✓			✓	✓	✓
	Negotiation Agent				✓	✓			✓
Service Providers	Service Architect	✓	✓	✓	✓	✓	✓	✓	✓
	Service Deployer				✓				
	Service Designer	✓		✓					
	Service Developer	✓	✓	✓	✓	✓			
	Service Provider				✓	✓	✓		✓
Supporting Roles	Manager	✓	✓	✓	✓	✓	✓	✓	✓
	Supporting Expert	✓	✓			✓	✓		
Coverage		84%	68%	42%	58%	53%	63%	48%	42%

Table 4.1: Coverage of the S-Cube Lifecycle by the Consolidated Stakeholder Types

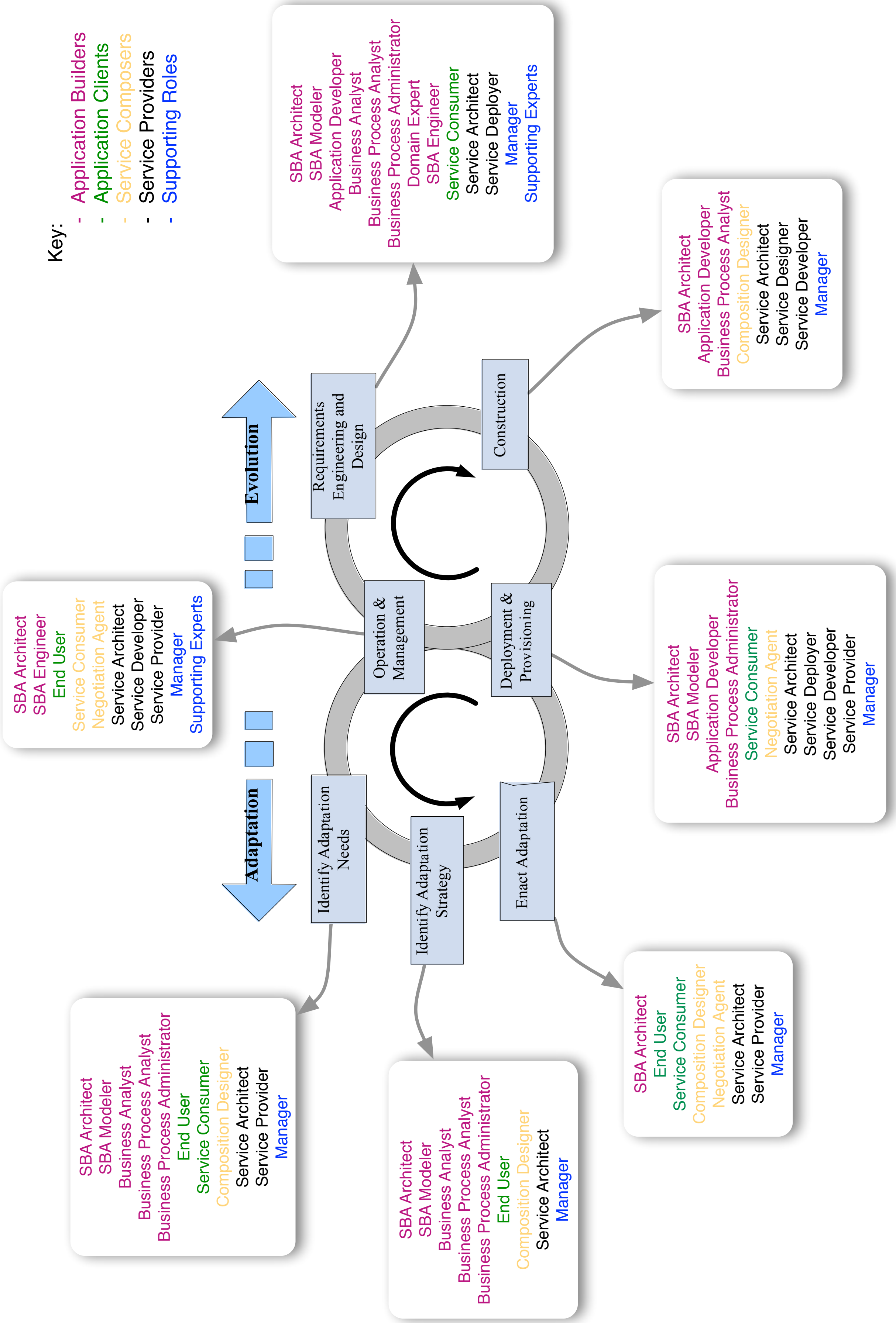

Figure 4.1: Consolidated Stakeholder Types Mapped To The S-Cube Lifecycle Phases

Chapter 5

Conclusions & Future Work

This deliverable has presented an initial set of stakeholder types present in the engineering, operation, adaptation and consumption of service-based applications. The motivation for performing this work is to understand which/what stakeholders are involved at each stage of the S-Cube service and service-based application lifecycle.

5.1 Future Work

Future work in this workpackage will be to continue along two paths: first, the work of Task T-IA-3.1.3 "Analysis of User Patterns and Methodologies" will refine the initial set of stakeholder types found in the process of completing this deliverable. It is suggested that rather than defining more stakeholders (which carries the risk of producing an unmanageable and incoherent set of stakeholder types), future work should concentrate on establishing the relationships between the stakeholder types found and refining their descriptions to produce focussed, crisp and clear definitions of each type. Secondly, the same task will define the user methodologies not included in this interim report. As we described at the start of Chapter 1, in this deliverable we have chosen to concentrate solely on the definition of stakeholders in this document and the methodologies will be presented in the follow-up deliverable to this interim deliverable, CD-IA-3.1.6 "Final Definition of Users Patterns & Methodologies".

As this is an integration activity, we will integrate the results of this deliverable into the S-Cube Knowledge Model (KM) being developed as part of workpackage WP-IA-1.1 — each of the stakeholder types will be added as a term within the KM and the descriptions from Appendix A will be used to provide the definitions necessary to complete it. As a KM term allows references and an indication of which S-Cube scenarios, industrial use-cases and research challenges are relevant to the term, each stakeholder type will also be integrated with the work of S-Cube's WP-IA-2.2 ("Alignment of European Industry Practices") and WP-IA-3.2 ("Integration Framework") activities that have defined several pilot use cases and scenarios respectively.

Appendix A

Stakeholder Specifications

The following are the stakeholder templates as collected from S-Cube participants in Step 1 of the initial collection methodology described in Section 3.1 (page 9). The contributions are listed in the standard S-Cube participant order.

Please note that some S-Cube participants submitted more than one definition for the same stakeholder type (e.g., City University have submitted two definitions for service developer). This is because more than one researcher in the same institution has sent in the same stakeholders with the same type. Each of the stakeholders is presented for completeness and this is why more than one type may exist from the same partner.

UniDUE-Essen

Direct User

Contributor	Andreas Metzger (UniDue)
Type of stakeholder	Direct User
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design, Operation and Management, Identify adaptation need.
Activities	Requirements elicitation, definition, and agreement.
Description	Direct users have specific requirements for the user interface of the system.
Peculiarity in service engineering	
Interactions with other stakeholders	Interact with all stakeholder roles relevant for service and system development, maintenance and evolution.
References	K. Pohl. Requirements Engineering: Principles, Techniques and Methods. Springer, 2010 (to be published).
Glossary	
Related challenges	

Indirect User

Contributor	Andreas Metzger (UniDue)
Type of stakeholder	Indirect User.
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design, Operation and Management, Identify adaptation need.
Activities	Requirements elicitation, definition, and agreement.
Description	Indirect users are, amongst others, persons and systems with the following characteristics: The person or system does not use the system himself (itself) but is indirectly involved in the usage; The person or system influences the usage of the system; The person or system benefits from the usage indirectly. For example, the head of a department is an indirect user of an information system if the employees of the department present cumulative reports generated by the information system to him.
Peculiarity in service engineering	
Interactions with other stakeholders	Interact with all stakeholder roles relevant for service and system development, maintenance and evolution.
References	K. Pohl. Requirements Engineering: Principles, Techniques and Methods. Springer, 2010 (to be published)
Glossary	
Related challenges	

Experts for User Interfaces

Contributor	Andreas Metzger (UniDue)
Type of stakeholder	Experts for User Interfaces.
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design.
Activities	Requirements elicitation, definition, and agreement.
Description	Experts for the user interface design know existing standards and laws applicable to the user interface to be designed as well as common pitfalls and obstacles faced when developing user interfaces of a certain type, thereby contributing to the system requirements. In addition, user interface experts can contribute significantly to the definition of usability tests.
Peculiarity in service engineering	
Interactions with other stakeholders	Interact with all stakeholder roles relevant for service and system development, maintenance and evolution.
References	K. Pohl. Requirements Engineering: Principles, Techniques and Methods. Springer, 2010 (to be published).
Glossary	
Related challenges	

Technology Consultants & Suppliers

Contributor	Osama Sammodi (UniDue)
Type of stakeholder	Technology Consultants and Suppliers.
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design, Operation and Management, Identify adaptation need
Activities	Requirements elicitation, definition, and agreement
Description	Technology consultants and suppliers are experts with knowledge about market trends and strategies regarding relevant software and hardware components, such as technology consultants or suppliers of software and hardware components.
Peculiarity in service engineering	
Interactions with other stakeholders	Interact with all stakeholder roles relevant for service and system development, maintenance and evolution.
References	K. Pohl. Requirements Engineering: Principles, Techniques and Methods. Springer, 2010 (to be published).
Glossary (S-Cube Knowledge Model)	
Related challenges	

Lawyers & Data Privacy Officers

Contributor	Osama Sammodi (UniDue)
Type of stakeholder	Lawyers & Data Privacy Officers.
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design.
Activities	Requirements elicitation, definition, and agreement
Description	Lawyers and data privacy officers have expert knowledge about what kinds of data can be stored in the system, how the data must be stored (e.g. with respect to encryption), what kinds of data must be anonymised, how long specific kinds of data can or must be stored, and after how long certain types of data can or must be deleted.
Peculiarity in service engineering	
Interactions with other stakeholders	Interact with all stakeholder roles relevant for service and system development, maintenance and evolution.
References	K. Pohl. Requirements Engineering: Principles, Techniques and Methods. Springer, 2010 (to be published).
Glossary	
Related challenges	

Domain Experts

Contributor	Osama Sammodi (UniDue)
Type of stakeholder	Domain Experts.
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design.
Activities	Requirements elicitation, definition, and agreement.
Description	Domain experts are an important type of requirement source in the subject facet. They provide relevant information for the system to be developed. They provide information about the subject domain of a system as well as the context objects (i.e., immaterial or material objects or persons which exist in the system context and need to be considered when defining the requirements for the system) about which the system must represent information.
Peculiarity in service engineering	
Interactions with other stakeholders	Interact with all stakeholder roles relevant for service and system development, maintenance and evolution.
References	K. Pohl. Requirements Engineering: Principles, Techniques and Methods. Springer, 2010 (to be published).
Glossary	
Related challenges	

Change Manager

Contributor	Osama Sammodi (UniDue)
Type of stakeholder	Change manager.
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design.
Activities	Requirements elicitation, definition, and agreement.
Description	The change manager is the manager of the change control board. In case of conflicts, the change manager tries to mediate between the parties involved. He is responsible for documenting the decisions made as well as for communicating the decisions and change integration activities to the corresponding stakeholders. Typically, the change manager is also responsible for monitoring change integration and reporting integration progress to the change control board. However, he may of course delegate parts of his responsibilities to other members of the change control board or other stakeholders.
Peculiarity in service engineering	
Interactions with other stakeholders	Interact with all stakeholder roles relevant for service and system development, maintenance and evolution.
References	K. Pohl. Requirements Engineering: Principles, Techniques and Methods. Springer, 2010 (to be published).
Glossary (S-Cube Knowledge Model)	
Related challenges	

Tilburg

Business Process Manager

Contributor	Rafiqul Haque (Tilburg)
Type of stakeholder	Business Manager (There are different types of business managers. Although, *Strategic Manager* plays the most crucial role in this regard but other types such as *Performance Manager* cannot be ignored. More specifically, defining requirements at this level is an aggregated effort by different types of managers. Thus, the role is titled as *Business Manager*).
S-Cube Lifecycle phases	Early Requirement Engineering, Requirements Engineering & Design, Operation and Management.
Activities	Requirement Definition, control business operations, monitor operations, optimize business process.
Description	• Responsible for defining business (process) requirements by abstracting processes. The requirements may cover multiple aspects such as business strategies, governing factors, and service policies that is composed of business rules. Notably, service policies are internal to an organization, which may not be disclosed publicly. These are required when an end-to-end system starts from the scratch or a new business relationship needs to be built. Inevitably, the governing factors of processes are in fact guarantees contained in a formal agreement between business participants. The very common guarantee types could be service quality guarantee that associates with service performance indicators and action guarantee. • Responsible for controlling business operations particularly making necessary changes in requirement specification. • Responsible for monitoring operations throughout the chain of active processes and identify bottlenecks such as special condition, erroneous operations, violation of agreements, and so on. • Responsible for optimization of existing business process by refining the process based on report generated after deploying process. The refining may trigger reengineering the existing requirements.
Peculiarity in service engineering	Interact enormously with the stakeholders external to the organization. The interactions may focus on quality aspects of services that a stakeholder such as service provider or service broker provides. The interaction helps to define the governing factors and all other requirements of services. Note that the “interaction” in this regard is not similar to the “interaction” in service composition mechanisms.

Interactions with other stakeholders	• Interact with *Service Provider* or *Service Broker* for defining governing factor of services.
S-Cube References	• S-Cube Authors. State of the art Survey on Business Process Modeling and Management, S-Cube Deliverable-CD-JRA 2.1.1, 2008.
Glossary	Business Process , Business Process Management.
Related challenges	Some of the existing engineering methodologies such as SDLC and SOMA incorporate BPM in their functional domain, hence, they can be deemed as good candidates but, it is unclear whether these methodologies are adequately supportive to all the activities involve in BPM lifecycle especially interactions with external stakeholder while engineering requirements for BPM applications (which is service based). The interactions are important in this regard since BPM applications are fully service centric. The methodologies require to add a means that underpins a clear visualization of interactions between the stakeholders.

Business Process Modeler

Contributor	Rafiqul Haque (Tilburg)
Type of stakeholder	Business Process Modeler.
S-Cube Lifecycle phases	Requirement Engineering and Designing, Construction.
Activities	Modeling business (executable) process.
Description	Responsible for modeling business processes capturing required services to reach organizational goal. Intrinsically, business process analysts use this process model.
Peculiarity in service engineering	Visualizes services as process activities in a business process model. This visualization facilitates analyzing and refining service based process model to optimize business process. Inevitably, business analyst performs the analytical activities.
Interactions with other stakeholders	Interact with *Business Process Analyst* to model business processes and also to refine (suggested by business analyst) the process models to increase the correctness taking requirements into account.
S-Cube References	• S-Cube Authors. State of the art Survey on Business Process Modeling and Management, S-Cube Deliverable-CD-JRA 2.1.1, 2008.
Glossary	Business Process, Business Process Model.
Related challenges	There is no gap found between this stakeholder specification and engineering methodologies owing to simplicity of this role concerning its jobtype. Existing methodologies are supportive to this role.

Business Process Analyst

Contributor	Rafiqul Haque
Type of stakeholder	Business Process Analyst.
S-Cube Lifecycle phases	Requirements Engineering & Design, Construction.
Activities	Investigate requirements through modeling, simulating and analyzing “As Is” and “To Be” process states.
Description	Dealing with the tactical aspects of BPM that is discovering, validating, documenting business process-related knowledge entailing business requirements, functional requirements, and non-functional requirements [S-Cube Authors, 2008] through modeling, simulating and analyzing current and future business process states. Business analysts play a vital role (in participation with business manager) in process optimization in particular, identifying certain activities in the process that need to be improved.
Peculiarity in service engineering	Business analysts interact enormously with stakeholders external to the organization because, they play a role as mediator between business and information technology. Analysts design the process based on requirements and interact with service brokers or service providers to make sure the stakeholders deliver required services that realize the business processes.
Interactions with other stakeholders	• Interact with *Service Provider* and *Service Broker* for the service requirement specifications. • Interact with *Business Process Architect* for service composition.
S-Cube References	S-Cube Authors. State of the art Survey on Business Process Modeling and Management, S-Cube Deliverable-CD-JRA 2.1.1, 2008.
Glossary	Business Process Management, Business Process, Business Process Design, Service Composition.
Related challenges	The gap between this stakeholder specification and methodological support is similar to the gap described for business manager. Thus, the suggestion concerning fill the gap is also the same. (See the stakeholder specification of business manager).

Business Process Architect / Integrator

Contributor	Rafiqul Haque (Tilburg).
Type of stakeholder	Business Process Architect and/or Integrator.
S-Cube Lifecycle phases	Construction.
Activities	Identify dependencies and interrelationship between processes, and compose services.
Description	Building both small scale as well as large scale service based application by composing services.
Peculiarity in service engineering	Compose services considering several factors including predefined business goal and service behaviors that are governed by agreement specification entails action guarantees as well as QoS guarantees. A service can be composed from existing services, thus, business process architect interacts numerously with *Business Analyst* who has clear notion of valid requirements for services.
Interactions with other stakeholders	Interact with *Business Analyst* for service composition.
S-Cube References	• S-Cube Authors. State of the art Survey on Business Process Modeling and Management, S-Cube Deliverable-CD-JRA 2.1.1, 2008. • S-Cube Authors. Separate Design Knowledge Model for Engineering and Service Based Computing, S-Cube Deliverable-CD-JRA 1.1.2, 2009.
Glossary	Business Process, Quality of Service, Service Composition, Service Based Application.
Related challenges	This role is plainly technical. Many of the existing engineering methodologies are adequately supportive to service composition. Thus, there is no substantial gap with existing methodologies.

City

Requirements Engineer

Contributor	City
Type of stakeholder	Requirements engineer
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design.
Activities	Requirements elicitation, requirements documentation.
Description	Responsible for the discovery, negotiation, and documentation of stakeholders requirements.
Peculiarity in service engineering	Limited interaction with end user.
Interactions with other stakeholders	Interaction with end users, service providers and developers for requirements elicitation, negotiation and documentation.
References	
Glossary	Requirements engineering.
Related challenges	Interaction with all involved stakeholders, especially consumers and end users, is rendered difficult by their lack of integration in service oriented engineering.

SBA Assembler

Contributor	City
Type of stakeholder	SBA assembler
S-Cube Lifecycle phases	Design, Construction.
Activities	Discovery and assembly/integration of services into SBAs according to required specifications.
Description	Service requestor looking to assemble the services into an SBA. Responsible for the discovery and integration of services fulfilling established requirements.
Peculiarity in service engineering	The role is peculiar to service engineering. Has to perform service discovery.
Interactions with other stakeholders	Possible interactions with developers or providers for discovery/ composition-related information if needed.
References	Lu, J. and Y. Yu (2007). "Web Service Search: Who, When, What, and How." Lecture Notes in Computer Science **4832**: 284.
Glossary	Service composition, service discovery.
Related challenges	Manual service discovery is a challenge due to lack of tool support for such stakeholders' information discovery behaviour (e.g. interaction with peers for recommendations), and lack of enough/ appropriate service description/information.

End-user

Contributor	City
Type of stakeholder	End-user
S-Cube Lifecycle phases	Requirement engineering and design, adaptation.
Activities	Use of the service/SBA to perform tasks.
Description	End user of the software, who interacts with it directly through a user interface or is the beneficiary/user of automated outputs.
Peculiarity in service engineering	Cannot always be foreseen or characterised.
Interactions with other stakeholders	Interaction with provider or broker to access to the service and possibly set service level agreements.
References	
Glossary	User experience; usability; SLA.
Related challenges	Variations in the user experience for the same SBA due to the fact that for a given SBA, the services used can be changing. Performing user centred design of SBAs for a better fit and efficiency of the software to the end user. Encouraging end-user's composition of their own SBAs.

Business Process Engineer

Contributor	City
Type of stakeholder	Business process engineer
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design.
Activities	Elicit business requirements, evaluate existing services' known attributes.
Description	Evaluate existing services to see if they meet business needs.
Peculiarity in service engineering	Operate on existing services with known specifications.
Interactions with other stakeholders	Passes data and feedback on to service developer.
References	M. Brian Blake, "Decomposing Composition: Service-Oriented Software Engineers," IEEE Software, vol. 24, no. 6, pp. 68-77, Nov./ Dec. 2007, doi:10.1109/MS.2007.162
Glossary	Requirements engineering, business process.
Related challenges	Basing the evaluation activities on existing services restricts the range of requirements and feedback that can be elicited and passed on to other phases of the development process.

Domain Expert

Contributor	City
Type of stakeholder	Domain expert
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design.
Activities	Express the system requirements.
Description	Model requirements.
Peculiarity in service engineering	The modelling constructs have to be defined so that different concepts and elements are distinguishable to facilitate service discovery.
Interactions with other stakeholders	Service designers and developers (for feedback on and handover of the requirements).
References	Daniel A. Menascé, John M. Ewing, Hassan Gomaa, Sam Malex, João P. Sousa, "A framework for utility-based service oriented design in SASSY" in Proc of the first joint WOSP/SIPEW international conference on Performance engineering San Jose, California, USA pp. 27-36, 2010.
Glossary	Requirements engineering.
Related challenges	Lacks the training or requirements engineer or software engineers for requirements elicitation.

Enterprise Architect

Contributor	City
Type of stakeholder	Enterprise Architect
S-Cube Lifecycle phases	(overseeing all, more directly involved in:) Requirements engineering and design Operation, management and quality assurance.
Activities	Help ensure that all the application architects are discussing their projects on a regular basis to identify services that they can expose or consume. Where an existing service cannot be reused, ensure that every opportunity to build a new service is exploited. Confirm that services are built on sound technology and are capable of meeting the established SLAs. Define mechanisms to measure and track SLAs. Define enterprise-wide SOA policies – regarding governance, security, disaster recovery etc. Key decision makers for solutions involving Web services management, orchestration, and Enterprise Service Bus.
Description	Defines SOA governing policies, best practices and procedures for each SBA; promotes and fosters the adoption of SOA with a focus on the application of technology to increase operational effectiveness and efficiency.

Peculiarity in service engineering	Additional responsibilities: as well as integrating a business plan with the technical capabilities as in a traditional IT context, responsible for developing an SOA framework and strategy and for ensuring an optimal use/ performance level of services.
Interactions with other stakeholders	application architects and developers – help them understand the fundamentals of SOA (where appropriate) and translate business requirements into meaningful services that they can implement and expose. Any other involved stakeholder, for coordination.
References	Ott, Christian and Korthaus, Axel and Böhmann, Tilo and Rosemann, Michael and Krcmar, Helmut (2010) ***Towards a reference model for SOA governance*** [online] http://eprints.qut.edu.au/31057/1/c31057.pdf (Working Paper) Kunal Mittal, **Create the ideal SOA team** [online] http://www.ibm.com/developerworks/ibm/library/ar-soateam/index.html
Glossary	Enterprise service bus, orchestration, governance, Architectural Knowledge, Architectural Knowledge Management.
Related challenges	Coordinating a broad range of activities and tasks throughout the lifecycle.

Application Architect

Contributor	City
Type of stakeholder	Application Architect.
S-Cube Lifecycle phases	Construction and quality assurance Operation, management and quality assurance.
Activities	Ensure that the code being written is service oriented and follows the agreed best practices and processes. Determine the level of granularity in what to expose as a service, and how to expose it. Ensure that opportunities for service reuse are exploited.
Description	Responsible for applications service-oriented architecture and ensuring that the architecture can support current and future business needs. Responsible for all the technical aspects of service delivery and consumption, including measurability of adherence to SLAs, meeting governance policies, and enforcing and ensuring security polices.
Peculiarity in service engineering	Architecture style used.
Interactions with other stakeholders	Enterprise architect, other other application architects, and development team (to ensure that the services are properly built, discovered, secured, used, and measured according to agreed standards and policies).
References	Ott, Christian and Korthaus, Axel and Böhmann, Tilo and Rosemann, Michael and Krcmar, Helmut (2010) ***Towards a reference model for SOA governance*** [online] http://eprints.qut.edu.au/31057/1/c31057.pdf (Working Paper) Kunal Mittal, **Create the ideal SOA team** [online] http://www.ibm.com/developerworks/ibm/library/ar-soateam/index.html

Glossary	Architectural Knowledge, Architectural Knowledge Management.
Related challenges	Translate business requirements into meaningful services without over-engineering the solutions.

Service Developer (1)

Contributor	City
Type of stakeholder	Service Developer.
S-Cube Lifecycle phases	Construction and quality assurance, Operation, management and quality assurance, Deployment and provisioning.
Activities	Develops service interface, implementation, and invocation code; appropriately build the services required to satisfy the business processes; enforce good design principles for error handling, tracking/auditing, data translation, and security, and make sure they are incorporated into any service code.
Description	Responsible for service development, evolution, and maintenance.
Peculiarity in service engineering	Software architecture and implementation technologies used; resulting differences in development lifecycle (length and approach) used.
Interactions with other stakeholders	SOA architects (see related templates); other service developers.
References	Kunal Mittal, **Create the ideal SOA team** [online] http://www.ibm.com/developerworks/ibm/library/ar-soateam/index.html
Glossary	Service based application construction.
Related challenges	Technical and organizational challenges related to service oriented software development.

Business Analyst

Contributor	City
Type of stakeholder	Business analyst.
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design.
Activities	Communicate with executives and users at a strategic level to understand their requirements for the system, and with the technical team members to transform the established requirements into technical specifications that can be implemented and tested.
Description	Responsible for providing domain knowledge, translating requirements into processes and services.
Peculiarity in service engineering	Has to think in terms of services, and to work with the technical team to design and build those services that are necessary and leverage those that already exist.
Interactions with other stakeholders	SOA architects, developers (including for the development of test cases).

References	Ott, Christian and Korthaus, Axel and Böhmann, Tilo and Rosemann, Michael and Krcmar, Helmut (2010) ***Towards a reference model for SOA governance*** [online] http://eprints.qut.edu.au/31057/1/c31057.pdf (Working Paper) Kunal Mittal, **Create the ideal SOA team** [online] http://www.ibm.com/developerworks/ibm/library/ar-soateam/index.html
Glossary	Business process, business process analysis, monitoring and auditing, business process reusability.
Related challenges	Understand and translate the language of both business users and providers. Identify and analyze services.

Project Manager

Contributor	City
Type of stakeholder	Project manager
S-Cube Lifecycle phases	Oversees all.
Activities	Define service-level agreements (SLAs) and ensure they are attainable; coordination and tracking of services and SBAs implementation and execution.
Description	Responsible for defining project plans, implementing the plans and monitoring the project as well as establishing the appropriate service-level agreements and resource usage.
Peculiarity in service engineering	Needs to plan for much shorter delivery cycles.
Interactions with other stakeholders	Business stakeholders, service consumers (e.g. to define SLAs); development team and architects for project coordination.
References	Ott, Christian and Korthaus, Axel and Böhmann, Tilo and Rosemann, Michael and Krcmar, Helmut (2010) ***Towards a reference model for SOA governance*** [online] http://eprints.qut.edu.au/31057/1/c31057.pdf (Working Paper) Kunal Mittal, **Create the ideal SOA team** [online] http://www.ibm.com/developerworks/ibm/library/ar-soateam/index.html
Glossary	Software life cycle model; service life cycle model.
Related challenges	Composing with shorter delivery cycles.

Service Designer

Contributor	City
Type of stakeholder	Service designer
S-Cube Lifecycle phases	Requirements Engineering & Design.
Activities	Decide on operations grouping, i.e., whether a single service is better expressed as a number of collaborating or independent services; on the naming of operations and services; on the granularity of service operations; on the reuse of existing services; on error handling and all other aspects that influence the quality of the service being designed.
Description	
Peculiarity in service engineering	Because XML forms the basis for documents exchanged between service consumers and service providers in most organizations, a service designer absolutely needs to have good, if not very good command of XML and related standards and technologies.
Interactions with other stakeholders	SOA architects, developers team.
References	
Glossary	Design for adaptation; design for monitoring; design principle; design for reuse; service design.
Related challenges	Assess services' suitability for the particular problem and re-use potential in a different or more general scenario.

Service Developer (2)

Contributor	Ricardo Contreras (City)
Type of stakeholder	Service Developer
S-Cube Lifecycle phases	Requirement Engineering & Design, Construction.
Activities	Rule creation: creation of patterns for rules and rules for monitoring web services behavior (in *Requirements Engineering*, *Design* and *Construction*.
Description	Responsible for the creation of monitoring rules.
Peculiarity in service engineering	Service developers need to take into consideration the outcome of monitoring activities to ameliorate their services.
Interactions with other stakeholders	Interact with service broker for service publishing and system analyst for user context characteristics.
References	A. Zisman, R. Contreras. *A Pattern-based approach for monitoring adaptation*, IEEE International Conference on Software-Science, Technology and Engineering (SwSTE 2010), June 15-16, Herzlia, Israel
Glossary	Service registry.
Related Challenges	Existing service engineering methodologies do not support the creation of patterns and rules for monitoring service behavior.

Service-Based System Builder

Contributor	Ricardo Contreras (City)
Type of stakeholder	Service-based system builder.
S-Cube Lifecycle phases	Construction, Adaptation Need, Adaptation Strategy, Enact Adaptation, Requirement Engineering & Design.
Activities	Patterns creation: specification of patterns for monitoring rules in *Requirement engineering, Design and Construction* Rules deployment: identification of monitoring rules in *Adaptation need*. Rules creation/modification: adaptation and creation of new monitoring rules in *Adaptation strategy* and *Enact adaptation*.
Description	Responsible for selection, modification and creation of new monitoring rules, and patterns for the monitoring rules.
Peculiarity in service engineering	Service-based system builders need to provide monitoring patterns to support monitoring adaptation.
Interactions with other stakeholders	Interact with service providers for behavioral specifications and service consumers for user context characteristics.
References	A. Zisman, R. Contreras. *A Pattern-based approach for monitoring adaptation*, IEEE International Conference on Software-Science, Technology and Engineering (SwSTE 2010), June 15-16, Herzlia, Israel.
Glossary	Service composition.
Related Challenges	Existing service engineering methodologies are not able to support dynamic identification, modification and execution of monitoring rules. One possible approach could be to create patterns for the monitoring rules to support the above activities.

System Analyst

Contributor	Ricardo Contreras (City)
Type of stakeholder	System Analyst.
S-Cube Lifecycle phases	Early Requirements, Requirements Engineering & Design.
Activities	Contextual factors identification: user characteristics specification in *Early requirements.* User modeling: create models of the users in *Requirements engineering and Design*.
Description	Responsible for identifying the characteristics of the users to be monitored when the users interact with the system.
Peculiarity in service engineering	System analyst interacts with the user to learn and document his/her characteristics and to provide way to model these characteristics.
Interactions with other stakeholders	Interact with service providers for behavioral specifications and service consumers for user context characteristics.
References	A. Zisman, R. Contreras. *A Pattern-based approach for monitoring adaptation*, IEEE International Conference on Software-Science, Technology and Engineering (SwSTE 2010), June 15-16, Herzlia, Israel.

Glossary	Service consumer.
Related Challenges	Existing service engineering methodologies do not take into consideration the human context factor, which might trigger a selection, modification or even creation of monitoring rules. A possible solution could be the creation of a user model representing user contexts. This will facilitate the identification of context changes and the identification and modification of monitoring rules.

Application Builder

Contributor	Khaled Mahbub (City)
Type of stakeholder	Application builder.
S-Cube Lifecycle phases	Early requirements engineering, Requirements engineering & design, Construction, Deployment & provisioning.
Activities	Requirements analysis (in /Early requirements engineering/ and /Requirements engineering & design/)[1][2][3]; Requirements Engineering - analysis of specification of existing services and new services (in /Requirements engineering & design/)[1][2][3]; Application design - architectural design of service based systems using existing and new services (in /Requirements engineering & design/ and /Construction/)[1][2][3]; Application implementation – development of service based system using existing and new services (in /Construction/)[1]; Application provision - deployment of service based system (in /Deployment & provisioning/)[1].
Description	Responsible for design and implementation of service based system using existing and newly developed services. Also responsible for the deployment of the developed service based system [1].
Peculiarity in service engineering	Application builder and service provider interact in the service specification and design level rather than in the requirements specification level which is common between service requestor (e.g. application builder) and service provider in traditional software engineering.
Interactions with other stakeholders	Interact with the developer of the existing services for service specification.

References	1. Khaled Mahbub and Andrea Zisman, "Replacement Policies for Service-Based Systems", 2nd International Workshop on Service Monitoring, Adaptation and Beyond (MONA+), Collocated with ICSOC/ServiceWave, Stockholm, Swedeen, November 23-24, 2009 2. Khaled Mahbub and George Spanoudakis, "Proactive SLA Negotiation for Service Based Systems", 4th International Workshop of Software Engineering for Adaptive Service-Oriented Systems, 2010, (submitted) 3. Khaled Mahbub and George Spanoudakis, "A Framework for Proactive SLA Negotiation", 5th International Conference on Software and Data Technologies, 2010, (submitted)
Glossary	Service process model, service based system, service orchestration.
Related challenges	Existing service engineering methodologies are not adequate to support validation of service based systems after dynamic adaptation (e.g. runtime replacement of a service in the composition). One possible solution could be history based validation, i.e. traces of previous successful transactions can be used to validate the composition after dynamic adaptation.

FBK

End User (of User-Centric SBAs)

Contributor	Raman Kazhamiakin (FBK)
Type of stakeholder	End user
S-Cube Lifecycle phases	Operation and Management Identify Adaptation Needs Identify Adaptation Strategy
Activities	During the Operation and Management phase the activities of the end user include - **Definition of personal preferences**. The definition of personal preferences and constraints that will be used through the other phases of the SBA life-cycle to drive the monitoring, adaptation, and provisioning activities. These preferences and constraints may characterize the user goals and requirements; define personal QoS constraints; preferences over the services and service types included in the service compositions [2]. - **Definition of user-specific rules**. Explicit definition of user-specific ways (e.g., through rules) to customize the way the services are related and integrated [1]. - **SBA execution**. Operations over the involved application functionalities and services that trigger service composition execution and adaptation [3]. That is, the user puts in his agenda an appointment that may require organization of a trip, hotel and train booking, etc. through the composition of available services. During the adaptation process, the end user actively participates to the process by - **Making decision on what kind of adaptation is necessary** (Identify Adaptation Needs phase). In the user-centric systems this decision is up to the user, while the role of the SBA is to enable and propose to the user possible alternatives by means of the available services and service compositions. - **Making decision on how the adaptation should be performed** (Identify Adaptation Strategy phase). That is, the user is explicitly (through the interaction with the SBA) or implicitly (through the preferences and constraints) decides what the concrete actions to be performed in order to adapt the current process are. For example, during the trip organization the user decides which types of transport to use, which company (service) to prefer, and what to do when certain problems occur requiring adaptation (e.g., retry, use another services, use alternative means, etc).

Description	User-centric services provide functionalities directly to the end-users and range from internet services (information services, booking services, online shopping, etc), to telco services and location-based services available through specific interfaces (RFID, QR-codes, etc). The user-centric SBAs aim at composing, harmonizing, providing, and executing those services in a way, where the services become part of the user's activities, constraints, and goals rather than remaining isolated. The end user of such an SBA plays the key role in the process of service composition provisioning and adaptation, as it is necessary to foster the user control on those activities requiring that no critical information is transmitted without involving the user, and no actions that are redundant or even harmful for the user are initiated by the platform autonomously. Therefore, such systems provide the means and capabilities for the user not only to activate and invoke the SBA functionalities and services, but to continuously control their execution and adaptation by making critical decision, selecting appropriate strategies and actions.
Peculiarity in service engineering	Differently from traditional systems, user-centric SBAs aim to integrate services that are completely independent, use heterogeneous formats and protocols, and discovered and exploited dynamically in different contexts and situations. One of the key challenges here is to provide means to associate those independent and heterogeneous services to the user-specific information thus providing a basis for their integration and composition [3]. This is specifically complex problem in case of highly dynamic scenarios, e.g., when the services become available/unavailable in changing context (e.g., SBAs delivered through mobile phones and devices, while the user is moving in different physical, social, and operational environments).
Interactions with other stakeholders	Interactions with other stakeholders happen through the study of the target user groups by collecting requirements and user feedback when proposing application mock-ups and prototypes and through the specific methodologies within the user-centered design approaches [4].

References	Related work 1. Jian Yu, Paolo Falcarin, Jose M. del Alamo, Juergen Sienel, Quan Z. Sheng, Jose F. Mejia, "A User-Centric Mobile Service Creation Approach Converging Telco and IT Services," Mobile Business, International Conference on, 2009, pp. 238-242 2. Mika Klemettinen, "Enabling Technologies for Mobile Services: The MobiLife Book", Wiley, 2007. 3. Raman Kazhamiakin, Piergiorgio Bertoli, Massimo Paolucci, Marco Pistore, Matthias Wagner: Having Services "YourWay!": Towards User-Centric Composition of Mobile Services. FIS 2008, pp. 94-106 4. Kurvinen, E., Aftelak, A., and Häyrynen, A. 2006. User-centered design in the context of large and distributed projects. In CHI '06 Extended Abstracts on Human Factors in Computing Systems.
Glossary	• Human-Computer Interaction • Personalization • Persona • User Experience • User-Centric SBA (to be added)
Related challenges	• Context- and HCI-aware SBA monitoring and adaptation • HCI and context aspects in the development of service based applications • Mixed initiative SBA adaptation

LERO

Business Analyst

Contributor	Stephen Lane (Lero)
Type of stakeholder	Business Analyst.
S-Cube Lifecycle phases	Identify adaptation needs, Identify Adaptation Strategies.
Activities	Define adaptation requirements, define monitoring requirements, define monitored property, design adaptation strategy.
Description	Interprets business rules and translates them in to system requirements. In the case of adaptable Service Based Applications (SBAs) the analyst has to determine when it is appropriate for an application to adapt by eliciting requirements from business stakeholders.
Peculiarity in service engineering	When adaptable SBAs are concerned the business analyst needs to be aware of alternative services that are available during runtime as well as which monitored events should trigger adaptation.
Interactions with other stakeholders	The business analyst needs to work closely with application developers while adaptable SBAs are being developed.
References	S. Lane and I. Richardson, "Process models for service based applications: A systematic literature review," to be submitted: Journal of IST, 2010.
Glossary	Service-Based Application, Adaptation, Life-Cycle Model.
Related challenges	There is no mention of this stakeholder in existing service model, and this should be addressed in future work.

SBA Developer / Service Consumer

Service Engineering Stakeholder Specification	
Contributor	Stephen Lane (Lero)
Type of stakeholder	SBA Developer/Service Consumer.
S-Cube Lifecycle phases	Identify adaptation strategy.
Activities	Provide monitoring functionality, Collect monitoring results for adaptation. Implement adaptation mechanism.
Description	In the context of the adaptation cycle of the S-Cube life-cycle the SBA Developer implements monitoring mechanisms that allow the SBA to adapt, the developer also need to implement the adaptation logic. The SBA developer may work on evolving the application as well as adapting it.
Peculiarity in service engineering	A SBA developer/service consumer is distinctly different from a traditional developer as they are exclusively dealing with the composition of available services rather then implementing the underlying functionality themselves.
Interactions with other stakeholders	The SBA Developer/Service Consumer needs to work closely with business analysts in order to implement the features required by the business.

References	S. Lane and I. Richardson, "Process models for service based applications: A systematic literature review," to be submitted: Journal of IST, 2010.
Glossary	Service-Based Application, Adaptation, Life-Cycle Model.
Related challenges	Service developers are mentioned in the literature but their interactions with other stakeholders aren't discussed.

Service Developer / Service Provider

Contributor	Stephen Lane (Lero)
Type of stakeholder	Service Developer/Service Provider.
S-Cube Lifecycle phases	Construction.
Activities	Develop services, publish services.
Description	In the context of the adaptation cycle of the S-Cube life-cycle the Service Developer/Provider develops services and makes them available for client applications, the services that they produce may be consumed by adaptable SBAs during runtime.
Peculiarity in service engineering	A Service developer/service provider is distinctly different from a traditional developer as they are developing services without necessarily knowing where they will end up being consumed.
Interactions with other stakeholders	The Service developer/service provider needs to work closely with business analysts in order to implement services required by the business.
References	S. Lane and I. Richardson, "Process models for service based applications: A systematic literature review," to be submitted: Journal of IST, 2010.
Glossary	Service-Based Application, Adaptation, Life-Cycle Model.
Related challenges	Service developers are mentioned in the literature but their interactions with other stakeholders aren't discussed.

Service Engineer

Contributor	Sajid Ibrahim Hashmi & Ita Richardson (Lero)
Type of stakeholder	Service Engineer.
S-Cube Lifecycle phases	Operation and Management.
Activities	Service level agreement, troubleshooting, managing adaptive infrastructure, optimizing and controlling web service infrastructure, monitoring run time environment for availability, accessibility, performance, error detection, resolution, and auditing.
Description	Responsible for quality assurance and management of service based applications.
Peculiarity in service engineering	In service engineering, stake holders, e.g. configuration managers have to pay special attention to the configuration issues. This level of attention is much higher than the traditional software engineering approaches because poor configuration management may lead to system related failures like failures of key services, and deficiency in performance and productivity. This interaction is among stakeholders within the organization throughout the service based system development activities.
Interactions with other stakeholders	Monitoring of the service state during its execution, i.e. interaction with the service consumer to keep track of the particular version of the service requested, and interaction with the service provider to get the requested service.
References	Sajid Ibrahim Hashmi, Stephen Lane, Dimka Karastoyanova, and Ita Richardson, "A CMMI Based Configuration Management Framework to Manage the Quality of Service Based Application", Submitted to EuroSPI 2010.
Glossary	Quality Definition, Negotiation, Assurance.
Related challenges	Does the level of interaction among different stakeholders improve the Configuration Management quality assurance process?

POLIMI

Application Designer/Application Developer

Contributor	Valentina Mazza (POLIMI)
Type of stakeholder	Application Designer/Application Developer.
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design, Construction.
Activities	Legacy re-engineering, Business modeling, requirements definition, requirements-based service discovery, architecture time service discovery, service-centric architecture and composition design.
Description	Responsible for application design and construction using services.
Peculiarity in service engineering	System Integrator builds the applications starting from existing services. Such services are often external to the organization to which the service integrator belongs. During the design phase he is able to define the services that will be used by the service based application.
Interactions with other stakeholders	Interact with service provider for service centric architecture and composition design; Interact with service provider for service negotiation.
References	Separate design knowledge models for software engineering and service based computing. Technical report, Deliverable CD-JRA-1.1.2, S-Cube Consortium, May 2009.
Glossary	Service based application design, construction.
Related challenges	Services identified during design phase may change or be unavailable during the life of the application. Definition of adaptation actions.

Service Developer/Service Designer

Contributor	Valentina Mazza (POLIMI)
Type of stakeholder	Service Developer/service designer
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design, Construction.
Activities	Legacy re-engineering, Business modeling, requirements definition, architecture time service discovery.
Description	Responsible for service design and implementation.
Peculiarity in service engineering	Extensively interact with stakeholders that are possibly external to the organization in order to deliver the desired service.
Interactions with other stakeholders	Interact service consumers for the development of the service.
References	Separate design knowledge models for software engineering and service based computing. Technical report, Deliverable CD-JRA-1.1.2, S-Cube Consortium, May 2009.

Glossary	
Related challenges	Service construction.

Service Provider

Contributor	Valentina Mazza (POLIMI)
Type of stakeholder	Service Provider.
S-Cube Lifecycle phases	
Activities	Owner of the service.
Description	Owns the service and guarantee a desired QoS level.
Peculiarity in service engineering	He is the owner of the service, and stipulates contracts (having the form of SLAs) with the users that want to access to the service.
Interactions with other stakeholders	He could be the service developer.
References	Separate design knowledge models for software engineering and service based computing. Technical report, Deliverable CD-JRA-1.1.2, S-Cube Consortium, May 2009.
Glossary	Service provider.
Related challenges	

Service Consumer

Contributor	Valentina Mazza (POLIMI)
Type of stakeholder	Service Consumer.
S-Cube Lifecycle phases	Operation and Management.
Activities	Service usage, invocation.
Description	Responsible for service usage.
Peculiarity in service engineering	Service Consumer is the end user of the service.
Interactions with other stakeholders	Interact with service developer and service provider for service specification; Interact with service provider for service negotiation.
References	Separate design knowledge models for software engineering and service based computing. Technical report, Deliverable CD-JRA-1.1.2, S-Cube Consortium, May 2009.
Glossary	Service consumer, invocation.
Related challenges	Service Consumer uses the service without owning physically it. When he stipulates a contract with the service provider he acquires only the possibility of using the service, moreover services could change or be unavailable without advertisement.

UoC

Composition Designer

Contributor	George Baryannis (UoC)
Type of stakeholder	Composition designer.
S-Cube Lifecycle phases	Construction, Adaptation.
Activities	Manual construction of a service composition, Identify adaptation strategy.
Description	Responsible for examining the available services and the goals stated in the requirements document and produce a composition schema (for example a BPEL process) that realizes these goals. Also responsible for defining adaptation strategies and how they modify the initial composition schema (design-time adaptation support).
Peculiarity in service engineering	The composition designer has to take into account that participating services may not always available and reflect this on the composition schema by providing adaptation mechanisms.
Interactions with other stakeholders	Interacts with stakeholders responsible for providing service specifications (e.g. service providers); Interacts with stakeholders responsible for providing the requirements document.
References	Deliverable PO-JRA-2.2.1 “Overview of the State of the Art in Composition and Coordination of Services”.
Glossary	Composition Schema, Service Composition, Formal Specification, Adaptation Strategy, Adaptation Mechanism, Process, Process Model, Service Process Model.
Related challenges	

Negotiation Agent

Contributor	Chrysostomos Zeginis (UoC)
Type of stakeholder	Negotiation Agent.
S-Cube Lifecycle phases	Service Level Agreement Negotiation.
Activities	Formulating, exchanging and evaluating a number of SLA Proposals in order to reach an SLA Contract for the provision/consumption of a service.
Description	Negotiating the SLA proposals on behalf of a service provider or a service consumer.
Peculiarity in service engineering	
Interactions with other stakeholders	Interacts on behalf of a service provider negotiating the SLA proposals. Interacts on behalf of a service consumer negotiating the SLA proposals.

References	Michael Parkin, Dean Kuo, John Brooke, A Framework & Negotiation Protocol for Service Contracts, Proceedings of the IEEE International Conference on Services Computing 2006, pp: 253 - 256.
Glossary	Service Level Agreement, Service Level Agreement Negotiation.
Related challenges	

Service Provider

Contributor	Chrysostomos Zeginis (UoC)
Type of stakeholder	Service Provider
S-Cube Lifecycle phases	Quality Assurance of SBAs.
Activities	Determine whether the current execution, preserves specified properties according to consumer's needs.
Description	Monitor services to ensure that SLAs or policies are met.
Peculiarity in service engineering	In SOAs monitoring can be used to observe the status of SBAs - as in traditional software engineering - and services.
Interactions with other stakeholders	Interacts with the Negotiation Agent to negotiate the SLA proposals of Service Consumer.
References	Deliverable PO-JRA-1.3.1 "Survey of quality related aspects relevant for SBAs".
Glossary	Monitor, Monitoring, Monitored Property.
Related challenges	The challenge is to combine the results of these approaches with engineering principles, techniques and methods, e.g. to achieve an automated adaptation of SBAs due to monitoring results or to closely align requirements with current service provision.

Service Consumer

Contributor	Chrysostomos Zeginis (UoC)
Type of stakeholder	Service consumer.
S-Cube Lifecycle phases	Quality Assurance of SBAs.
Activities	Determine whether the current execution preserves specified properties according to consumer's needs.
Description	Monitor services to ensure SLAs or policies are met.
Peculiarity in service engineering	In SOAs monitoring can be used to observe the status of SBAs - as in traditional software engineering - and services.
Interactions with other stakeholders	Interacts with the Negotiation Agent to negotiate the SLA proposals of Service provider.
References	Deliverable PO-JRA-1.3.1 "Survey of quality related aspects relevant for SBAs".

Glossary	Monitor, Monitoring, Monitored Property.
Related challenges	The challenge is to combine the results of these approaches with engineering principles, techniques and methods, e.g. to achieve an automated adaptation of SBAs due to monitoring results or to closely align requirements with current service provision.

UPM

Service Designer

Contributor	Dragan Ivanovic, Manuel Carro (UPM)
Type of stakeholder	Service Designer.
S-Cube Lifecycle phases	Construction, Requirements Engineering & Design.
Activities	Re-engineering, process modeling, service specification, composition modeling and design, QoS requirement specification, QoS engineering for service compositions.
Description	A service designer is primarily responsible for designing (or choosing) the interface, message types, interaction patterns for a service, as well as for providing its abstract (logical) and concrete (executable) design. Usually, the designer is also at least partially involved in the previous stages of the service lifecycle, such as requirements engineering.
Peculiarity in service engineering	The service designer is supposed to have good knowledge of service technologies and standards (such as WSDL, XML, XML Schema, modeling formalisms and languages, WS-Policy and SLA), as well as of the SOA development policies and guidelines in the respective organizational setting (also known as the SOA Governance). The former include rules on a particular design methodology, development and usage of standard data dictionaries (based on internal and external document and message structures), usage of the standard and introduction of new service interfaces (to ensure interoperability), organization and maintenance of design/implementation repositories, service versioning, and service reuse.
Interactions with other stakeholders	• SOA domain architect for the key lifecycle decisions; • service provider for QoS parameters of service provision; • business and technical service owners for functional and non-functional properties of details of business processes and technical services.

References	David Booth, Hugo Haas, Francis McCabe, Eds. Web Services Architecture: W3C Working Group Note 11 February 2004. World Wide Web Consortium, 2004. Retrieved 2010-04-14 from: http://www.w3.org/TR/ws-arch/ Dragan Ivanovic, Manuel Carro, Manuel Hermenegildo. An Initial Proposal for Data-Aware Resource Analysis of Orchestrations with Applications to Predictive Monitoring. Proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+). 2009. To appear. Dragan Ivanović, Manuel Carro, Manuel Hermenegildo. Towards Data-Aware QoS-Driven Adaptation for Service Orchestrations. Technical Report CLIP5/2009.1, School of Computer Science, Technical University of Madrid. 2009. Stefan Tilkov. Roles in SOA Governance. InfoQ, 2007. Retrieved 2010-04-14 from: http://www.infoq.com/articles/tilkov-soa-roles
Glossary	Service Engineering, Service Design, QoS, Data-Awareness.
Related challenges	Current methodologies do not offer good support to service designers for analyzing computational cost and QoS bounds for individual services, as well as for refactoring/adaptation along these lines. Development of such tools and techniques is required.

Service Composition Designer

Contributor	Dragan Ivanovic, Manuel Carro (UPM)
Type of stakeholder	Service Composition Designer.
S-Cube Lifecycle phases	Construction, Requirements Engineering & Design.
Activities	Re-engineering, process modeling, service specification, composition modeling and design, QoS requirement specification, QoS engineering for service compositions.
Description	Service composition designer is a specialization of the role of service designer. Besides other service designer responsibilities, a service composition designer additionally has to specify the composition using an appropriate modeling/executable language, identify composition components, map the composition-level requirements to component constraints, and take into account possible impact of component behavior on the behavior of the entire composition. This is especially important for flexible, adaptable, QoS-aware service compositions, where both the design and runtime environment can benefit from inferring bounds for QoS attributes of the composition based on its structure and the QoS bounds of its components, to provide predictive monitoring and adaptation capabilities.
Peculiarity in service engineering	This particular role does not exist in the traditional software engineering process.

Interactions with other stakeholders	SOA domain architect for the key lifecycle decisions; service provider for QoS parameters of service provision; business and technical service owners for functional and non-functional properties of details of business processes and technical services.
References	David Booth, Hugo Haas, Francis McCabe, Eds. Web Services Architecture: W3C Working Group Note 11 February 2004. World Wide Web Consortium, 2004. Retrieved 2010-04-14 from: http://www.w3.org/TR/ws-arch/ Dragan Ivanovic, Manuel Carro, Manuel Hermenegildo. An Initial Proposal for Data-Aware Resource Analysis of Orchestrations with Applications to Predictive Monitoring. Proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+). 2009. To appear. Dragan Ivanović, Manuel Carro, Manuel Hermenegildo. Towards Data-Aware QoS-Driven Adaptation for Service Orchestrations. Technical Report CLIP5/2009.1, School of Computer Science, Technical University of Madrid. 2009. Stefan Tilkov. Roles in SOA Governance. InfoQ, 2007. Retrieved 2010-04-14 from: http://www.infoq.com/articles/tilkov-soa-roles
Glossary	Service Engineering, Service Design, QoS, Data-Awareness, Business Process Modeling, Service Composition
Related challenges	Current methodologies do not offer good support to service designers for analyzing computational cost and QoS bounds for individual services and their compositions, as well as for refactoring/adaptation along these lines. Development of such tools and techniques is required.

SOA Domain Architect

Contributor	Dragan Ivanovic, Manuel Carro (UPM)
Type of stakeholder	SOA Domain Architect.
S-Cube Lifecycle phases	(All)
Activities	Coordinating design and other lifecycle activities, ensuring standards compliance, resolving conflicts, ensuring business value of the business-domain SOA components.

Description	The job of a SOA domain architect is to coordinate design and development efforts of several service designers so as to ensure that these efforts result in a coherent, high-quality, enterprise-wide SOA, which delivers business value to the enterprise. The word "domain" denotes the business domain of the enterprise that SOA intends to support. In most organizations, services are developed by a number of system designers working concurrently or over a period of time. To ensure overall coherence, one lead designer is trusted with the role of SOA domain architect, with or without doing regular designer work in parallel.
Peculiarity in service engineering	The domain architect resolves conflicts that may arise between different design projects with respect to rules, policies and standards that are used in the entire organization, and advises on major decisions that drive the service lifecycle model, such as initiating requirements engineering, design, implementation, and deployment.
Interactions with other stakeholders	service designers to resolve conflicts and ensure coherence and consistency across different development efforts; business service owners to ensure that SOA delivers relevant business value.
References	David Booth, Hugo Haas, Francis McCabe, Eds. Web Services Architecture: W3C Working Group Note 11 February 2004. World Wide Web Consortium, 2004. Retrieved 2010-04-14 from: http://www.w3.org/TR/ws-arch/ Stefan Tilkov. Roles in SOA Governance. InfoQ, 2007. Retrieved 2010-04-14 from: http://www.infoq.com/articles/tilkov-soa-roles
Glossary	Business Value, Service Engineering, Service Life Cycle.
Related challenges	To properly view the role and interactions of this stakeholder in the development of service-oriented architecture, it should be put into an appropriate SOA Governance context.

SOA Platform Architect

Contributor	Dragan Ivanovic, Manuel Carro (UPM)
Type of stakeholder	SOA Platform Architect.
S-Cube Lifecycle phases	Operations Management, Deployment and Provisioning, Requirements & Construction for technical services (optionally)
Activities	Choice of technology options for the service platform. Coordinating design and other lifecycle activities, ensuring standards compliance, resolving conflicts, ensuring business value of the technical (enabling) SOA components.

Description	The major responsibility of an SOA platform architect is the choice of technologies for provision of services in a manner that maximizes interoperability, manageability and business value of the SOA platform. Flexibility is usually subordinated to these goals, and designers are typically given a choice of several configurations of security, coordination, enactment, discovery, monitoring, etc., infrastructures to chose from.
Peculiarity in service engineering	Platform architect does the domain architect's job when it comes to development of technical, or enabling, services for the enterprise platform (typically including those that perform authentication, security clearance, logging, coordination, etc.).
Interactions with other stakeholders	technical service owner on infrastructure policies; service designers to resolve conflicts and ensure coherence and consistency across different development efforts, when it comes to technical services.
References	David Booth, Hugo Haas, Francis McCabe, Eds. Web Services Architecture: W3C Working Group Note 11 February 2004. World Wide Web Consortium, 2004. Retrieved 2010-04-14 from: http://www.w3.org/TR/ws-arch/ Dragan Ivanovic, Manuel Carro, Manuel Hermenegildo. An Initial Proposal for Data-Aware Resource Analysis of Orchestrations with Applications to Predictive Monitoring. Proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+). 2009. To appear. Dragan Ivanović, Manuel Carro, Manuel Hermenegildo. Towards Data-Aware QoS-Driven Adaptation for Service Orchestrations. Technical Report CLIP5/2009.1, School of Computer Science, Technical University of Madrid. 2009. Stefan Tilkov. Roles in SOA Governance. InfoQ, 2007. Retrieved 2010-04-14 from: http://www.infoq.com/articles/tilkov-soa-roles
Glossary	Service infrastructure
Related challenges	New tools and methodologies need to be developed to ensure better interaction between platform architects and service designers/domain architects when it comes to data-aware QoS-driven prediction, proactive monitoring, and adaptation.

USTUTT

Service Deployer

Contributor	Alexander Nowak, Branimir Wetzstein (USTUTT)
Type of stakeholder	Service Deployer.
S-Cube Lifecycle phases	Deployment and Provisioning.
Activities	Deployment: • Configures and deploys service composition and services on the corresponding service middleware platform. Provisioning: • Provisions IT infrastructure.
Description	Configures and deploys service composition and services on the corresponding service middleware platform and IT infrastructure.
Peculiarity in service engineering	SLA-aware deployment.
Interactions with other stakeholders	Interaction with service developer who provides the service (composition) artifacts. Interaction with service architect who provides requirements considering deployment and provisioning (e.g., SLAs such as availability requirements, cost)
References	
Glossary	Service deployment, Quality of service, Service runtime
Related challenges	

Service Network Modeler

Contributor	Olha Danylevych (USTUTT)
Type of stakeholder	Service Network Modeler
S-Cube Lifecycle phases	Early Requirements Engineering, Identify Adaptation Needs, Identify Adaptation Strategy.

Activities	Early requirements Engineering: • Top-down modeling of the service networks: identification of (1) the participants that will be involved in the service network, (2) the service offerings and requests and providing (exchanges of services between the participants) and (3) the contracts associated with the providing (e.g. SLA, End User License Agreements); • Bottom-up creation of service network model: create service network models using information extracted from existing process models, service compositions and agreements, contracts and licenses among participants; • Specification of functional and not-functional requirements of the provided services, i.e. which functionality should the process/service composition realize and with which non-functional properties should if satisfy e.g. costs, duration. Identify Adaptation Needs and Steps in the Service Network based on e.g. the results of monitoring and KPI violations: • Changes of the partners in the network, changes in the terms of the contracts etc; • Identification the business processes affected by the changes and delegate the changes to the corresponding roles (Business Process Modelers and Owners).
Description	Manages the service network model and provides both functional and non-functional requirements that the underlying processes and service compositions have to satisfy.
Peculiarity in service engineering	
Interactions with other stakeholders	Interacts with Business Process Analysts and Owners who provide the requirements.
References	Bitsaki, Marina; Danylevych, Olha; van den Heuvel, Willem-Jan; Koutras, George; Leymann, Frank; Mancioppi, Michele; Nikolaou, Christos; Papazoglou, Mike: An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network. In: Petri, Mähönen (Hrsg); Klaus, Pohl (Hrsg); Thierry, Priol (Hrsg): Towards a Service-Based Internet, First European Conference, ServiceWave 2008.
Glossary	Service network, Adaptation Requirements and Objectives, Adaptation Strategy, Business Process, Business Process Modeling.
Related challenges	

Business Process Analyst

Contributor	Alexander Nowak, Branimir Wetzstein (USTUTT)
Type of stakeholder	Business Process Analyst.
S-Cube Lifecycle phases	Early Requirements Engineering, Requirements Engineering & Design Identify Adaptation Needs, Identify Adaptation Strategy.
Activities	Early requirements Engineering: • Specification of functional requirements (which functionality should the process realize?) of the business process (after analysis of existing processes, or within business process reengineering initiatives); • Specification of non-functional requirements (in respect to duration, cost, and quality of the process incl. target values (KPIs)). Requirements Engineering & Design: • Business process modeling (creating a process model which realizes those requirements, e.g., in BPMN); • Design for monitoring (specifying which KPIs are to be monitored based on the process model, and notifications/ alarms to be sent in case of KPI target violations). Identify Adaptation Needs: • Analyzes the causes of KPI violations (need for adaptation), and defines adaptation requirements. Identify Adaptation Strategy: • Develops an adaptation strategy based on adaptation requirements.
Description	Defines both functional and non-functional requirements and specifies then a business process model which covers those requirements.
Peculiarity in service engineering	
Interactions with other stakeholders	Interacts with Business Process Owner for specification of requirements. Interacts with Service architect by providing him the business process model and KPI definitions. Interacts with Business process owner and Service architect for identification of adaptation needs and development of an appropriate adaptation strategy.
References	Raman Kazhamiakin, Branimir Wetzstein, Dimka Karastoyanova, Marco Pistore and Frank Leymann. Adaptation of Service-Based Applications Based on Process Quality Factor Analysis. In Proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+), 2009. (http://bibadmin.s-cube-network.eu/show.php?id=275)
Glossary	Adaptation Requirements and Objectives, Adaptation Strategy, Business Process, Business Process Modeling, Business Activity Monitoring, Design for Monitoring, Key Performance Indicator.
Related challenges	

Business Process Owner

Contributor	Alexander Nowak, Branimir Wetzstein (USTUTT)
Type of stakeholder	Business Process Owner.
S-Cube Lifecycle phases	Early Requirements Engineering, Operation and Management Identify Adaptation Needs, Identify Adaptation Strategy.
Activities	Early requirements Engineering: • Specification of functional requirements (which functionality should the process realize?), providing process knowledge to business analyst for process optimization purposes; • Specification of non-functional requirements (in respect to duration, cost, and quality of the process incl. target values (KPIs)). Operation and Management: • Monitoring of KPIs (dashboards); Identify Adaptation Needs: • Analyses the causes of KPI violations (need for adaptation) and defines adaptation requirements. Identify Adaptation Strategy: • Develops an adaptation strategy based on adaptation requirements.
Description	Provides process knowledge and defines functional and non-functional requirements.
Peculiarity in service engineering	
Interactions with other stakeholders	Interacts with Business Process Analyst providing process knowledge, process optimization goals (e.g., KPI targets) etc. Interacts with Business Process Analyst and Service Architect for identification of adaptation needs and adaptation strategies.
References	Raman Kazhamiakin, Branimir Wetzstein, Dimka Karastoyanova, Marco Pistore and Frank Leymann. Adaptation of Service-Based Applications Based on Process Quality Factor Analysis. In Proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+), 2009. (http://bibadmin.s-cube-network.eu/show.php?id=275)
Glossary	Adaptation Requirements and Objectives, Adaptation Strategy, Business Process, Business Process Modeling, Business Activity Monitoring, Design for Monitoring, Key Performance Indicator.
Related challenges	

Service Architect

Contributor	Alexander Nowak, Branimir Wetzstein (USTUTT)
Type of stakeholder	Service Architect.
S-Cube Lifecycle phases	Requirements Engineering & Design, Construction, Identify Adaptation Needs, Identify Adaptation Strategy.
Activities	Requirements Engineering & Design: • Requirements-based service discovery: reuse existing services for implementation of business process activities; • Service specification: if reuse not possible, specify new services which are to be implemented by service developer. Construction: • Service composition design: IT Refinement, transforming the business process model to executable service composition (e.g., in BPEL); • Design for monitoring and specification of KPIs. Identify Adaptation Needs: • Analyzes the causes of KPI violations (need for adaptation), and defines together with business analyst the adaptation requirements. Identify Adaptation Strategy: • Develops with business analyst an adaptation strategy, in particular considering IT-level concerns (e.g., IT infrastructure performance optimization, service substitution etc.).
Description	Responsible for realization of business processes in terms of executable service compositions.
Peculiarity in service engineering	Service architect implements the business process by either reusing existing services or specifying new services which are to be implemented by the service developer.
Interactions with other stakeholders	Interacts with business analyst who provides the business process model and requirements and for identifying an adaptation strategy Interacts with service developer for implementation of new services needed in the business process.
References	Raman Kazhamiakin, Branimir Wetzstein, Dimka Karastoyanova, Marco Pistore and Frank Leymann. Adaptation of Service-Based Applications Based on Process Quality Factor Analysis. In Proceedings of the 2nd Workshop on Monitoring, Adaptation and Beyond (MONA+), 2009. (http://bibadmin.s-cube-network.eu/show.php?id=275)
Glossary	Adaptation Requirements and Objectives, Adaptation Strategy, Business Process, Service Composition, Service Discovery, Business Activity Monitoring, Design for Monitoring, Key Performance Indicator.
Related challenges	

Bibliography

[1] Qing Gu and Patricia Lago. A Stakeholder-Driven Service Life Cycle Model for SOA. In *Proceedings of the 2nd International Workshop on Service-Oriented Software Engineering (IW-SOSWE'07)*, pages 1–7, Dubrovnik, Croatia, 2007.

[2] S-Cube Network Participants. Annex I — "Description of Work", November 2007.

[3] S-Cube Partners. Separate Design Knowledge Models for Software Engineering & Service-Based Computing. Contractual Deliverable CD-JRA-1.1.2, S-Cube Network of Excellence, December 2008.

[4] S-Cube Partners. Initial Definition of Interfaces Between Layers. Project Objective PO-IA-3.1.2, S-Cube Network of Excellence, June 2009.

[5] S-Cube Partners. Integration Framework Baseline. Contractual Deliverable CD-IA-3.1.1, S-Cube Network of Excellence, March 2009.

[6] S-Cube Partners. First Version of Integration Framework. Contractual Deliverable, S-Cube Network of Excellence, December 2010.