SERVIAM Maintenance Framework

Mira Kajko-Mattsson, Michal Tepczynski

SERVIAM
Department of Computer and Systems Sciences
Stockholm University & Royal Institute of Technology, Sweden
mira@dsv.su.se, michalt@kth.se

Abstract

Web services systems impose additional complexity upon software maintenance and evolution processes. To handle it, we propose a framework for evolving and maintaining Web service systems. Our framework includes organisation, role, and process changes. The feedback on its credibility has been provided by ten software organisations in Poland and Sweden.

1. Introduction

Many times, advantages from technological innovations are gained at the expense of undermining the well-established software engineering methodologies. Examples from real-life may be found many. For instance, when going over to relational databases or object-orientation, we had to revise our software engineering methodologies and adapt them to manage the new technology products. We also encountered many unforeseen difficulties and problems related to the new technology. Presently, we should expect similar obstacles when adapting to Service Oriented Architecture (SOA), based on the Web service (WS) technologies.

It is not always easy to identify all possible changes to the methodologies at the early technology conception. However, one may facilitate the identification process by studying the differences between the traditional old technology software systems and the new technology ones. It is only then one may start creating models for managing the new technology software systems.

From the evolution and maintenance perspective, there are many things that have to be examined. These include the evolution and maintenance processes, products and roles involved within these processes, and the organisational changes required for transferring to the Web service mode.

Presently, many software organisations are in the process of introducing Web services. At this initial phase, they conduct various pilot studies during which they test the new technology [5]. Soon however, they will have to figure out how to manage their large-scale production. Hence, they need to know how to adapt their existing software engineering methodologies and organisational infrastructures for handling the new technology systems.

Unfortunately today, the organisations do not possess enough knowledge of how to evolve and maintain Web service systems. To aid them, we have created SERVIAM Maintenance Framework, a general framework for the evolution and maintenance of Web services. In this framework, we have pointed out matters unique for the evolution and maintenance of Web services and changes to be implemented to the traditional processes, organisations, and roles. The feedback on our framework’s credibility was elicited within ten software organisations in Poland and Sweden.

The remainder of this paper is the following. Section 2 shortly describes the methodology taken when constructing the framework. Section 3 presents matters unique for evolving and maintaining Web services. Sections 4 and 5 present our framework and the industrial feedback. Finally, Section 6 makes conclusions and suggestions for future work.

2. Methodology

When building the framework, we conducted the following steps:

- **Literature Study:** During the first step, we made a comprehensive literature study to find issues concerning the evolution and maintenance of Web services. Unfortunately, we found nothing of interest.
- **Panel Debate:** Due to the fact that nothing has been published about the subject, the lead author of this paper organised a panel on the IEEE conference [8]. The panel constituted our starting point for eliciting knowledge necessary for building our framework.
- **Identification of unique matters:** When eliciting the unique characteristics of Web service systems and procedures for evolving and maintaining them, we used various sources of information, such as the panel results.
and different publications providing feedback on the SOA systems based on Web service technologies and the new business model supported by them.

- **Framework Construction**: The unique characteristics were then used for constructing the framework.
- **Framework Evaluation**: We conducted semi-structured open-ended interviews using the questionnaire presented in Table 1. Our interviewees were both maintainers and their managers belonging to ten companies in Poland and Sweden. These companies were in the process of introducing Web services. Hence, they could provide us relevant feedback on the design of our framework. The organisations in Sweden were Volvo IT, Personec AB (part of TietoEnator), Scandinavian Airline Systems (SAS), Skandinaviska Enskilda Banken (SEB), Centrala Studiestödsnämnden (CSN), AMF Pension, Stockholm County Council and Swedish National Tax Board. The organisations in Poland were Prokom Software SA and ATENA Usugi Informatyczne i Finansowe Sp. z o.o (ATENA).

### 3. Unique Matters

The characteristics unique for evolving and maintaining Web services systems originate from the distinguishing features in the SOA architecture, their supporting WS technologies and the new business model. For this reason, we first describe them in Sections 3.1 and 3.2. We then describe the unique matters in Sections 3.3 – 3.6. They are presented from the product, organisational, role and process perspectives. Due to space restrictions, we can only scratch the subject on the surface. Interested readers are welcome to read [11].

#### 3.1. SOA and Web Services

An SOA is a style of design covering all aspects of creating and using business services throughout their life cycle. One possible way to achieve SOA is to use Web service technology supporting the self-contained, self-describing, and modular units that can be published, invoked and executed across the network [4]. Web service units are loosely-coupled, implemented in different languages, and, executed on different platforms [18, 19]. Together, they constitute a Web service software system.

Web service units communicate with one another through well-defined interfaces. They can be used as building blocks in different applications. Their fine-grained structure allows breaking down the architecture into a set of units that can be easily changed and reused.

The Web service technology promotes higher level of automation, and thereby it results in a substantial decrease of manual work. It provides a standardized and unified way of integrating all kinds of systems, from legacy mainframes, to mobile clients, to internal services, and to systems provided by external business partners. Their loose coupling leads to greater flexibility. The implementation of individual components can be changed without any harm to other parts of the system as long as the interfaces remain unmodified [2].

#### 3.2. New Business Model

Web services can be offered to a wide range of users coming from various organisations, thus providing great cross-vendor interoperability via a standardised integration mechanism, and broad external and internal integration capabilities [15]. They have significantly influenced models of modern businesses. The borders of business organizations and business systems supporting them have moved towards the outer world.

Potentially, Web services provide a capability to bind services at runtime, during execution. Organizations can pay for the services when required and used [1]. They do not necessarily need to own them. This introduces a new concept of a service on the software market. As within the rest of the general service market, this concept affects the notion of ownership. It implies that certain functionality can be used, but need not be necessarily owned [1]. It is paid for only when needed and used.

The service concept together with the technological capabilities to realise it have contributed to the creation of a new business model for using, promoting and selling software applications. This model provides better opportunities for more global and dynamic business collaboration among organisations. It encourages wide software integration, procurement and use/provision of external/internal components.

Modern businesses can now run across many collaborating parties. These parties do not always need to know one another and their collaboration may be initiated dynamically “on the fly”. They utilise the services that have been published, over the network in a common service marketplace, where one can find and procure required services. This leads to a multi-customer business model implying that services can be used by many service consumers at a time regardless of their location.

### Table 1. Our Questionnaire

<table>
<thead>
<tr>
<th>Introductory questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q.1: Do you support (i.e. evolve and maintain) Web services?</td>
</tr>
<tr>
<td>Q.2: How many Web services do you support?</td>
</tr>
<tr>
<td>Q.3: How many of them do come from the external providers?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Organisational level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q.4: Do you have a separate team responsible for the Web service management?</td>
</tr>
<tr>
<td>Q.5: Do you limit the collaboration to only the primary contractors?</td>
</tr>
<tr>
<td>Q.6: Do you collaborate with sub-contractors providing the lower-level services?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Role level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q.7: Have you designated roles responsible for supporting Web services?</td>
</tr>
<tr>
<td>Q.8: What are these roles and their responsibilities?</td>
</tr>
<tr>
<td>Q.9: What are the criteria of their assignments, i.e. modules, sub-systems, features, business processes, Web services?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Process level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q.8: Have you defined an SLA process?</td>
</tr>
<tr>
<td>Q.9: Do you include the management of Web services in the SLA documents?</td>
</tr>
<tr>
<td>Q.10: What issues do you address?</td>
</tr>
<tr>
<td>Q.11: Do you distinguish between critical and non-critical WSs in the SLAs?</td>
</tr>
<tr>
<td>Q.12: Have you defined a common evolution and maintenance process?</td>
</tr>
<tr>
<td>Q.13: How do you configure Web services?</td>
</tr>
</tbody>
</table>
By means of Web services, business organizations will become more global, and this will in turn, make the evolution and maintenance of supporting their software systems more global as well. The new business model will significantly change the way one views maintenance organizations, from single independent organisational units to a collection of distributed, cooperating and dependent ones. This will cost additional resources to manage the communication effort.

3.5. Role Perspective

Today, the traditional support roles are mainly assigned to particular systems, subsystems, or modules with clearly defined ownership, responsibilities, borders and relationships to other cooperating roles.

In the context of Web services, the existing traditional team structure is inefficient. Web services promote the creation of service supply networks, where final functionality can be composed out of many sub-services. There are no clear hierarchic boundaries between systems and sub-systems. Particular services can be shared and composed out of many other services, thus creating a complex network with many interdependencies. Moreover, these services are not always owned; they can be used without being owned. So, it is not clear who is the owner of these components.

The existing roles and responsibilities have to be changed in order to efficiently handle the evolution and maintenance activities. The nature of Web services will also put requirements on the competence development of all the roles involved. Good knowledge of the new continuously changing technology and of the continuously changing businesses will be required.

3.6. Process Perspective

Almost all software processes will be affected by the new technology and the new business model. The main reasons are greater reuse of components, higher business volatility, and limited process visibility.

Greater reuse of components accelerates the development, evolution and maintenance processes. Old components may be easily replaced with new ones as long as their interfaces stay intact. New solutions can be implemented faster, cheaper and with lower risks [2].

Higher volatility of today’s businesses, on the other hand, changes the dynamics and complexity of the evolution and maintenance processes. The multi-customer model and the fine-granularity of Web services will lead to more, often contradicting, change requests.

Limited insight into the evolution and maintenance processes is due to the fact that these processes are beyond the control of one organization. The evolution and maintenance activities are dependent on the collaboration...
3.6.2. Problem management. Changes mainly arise as a result of recognized problems or new business needs. When adding new features in the Web service systems, the engineers (consumers of Web services) will search for appropriate services on the service market, instead of developing them on their own. This will affect their requirements specification, testing, and impact analysis processes. The engineers will also have to consider the fact that the Web services offered on the service market today may be changed tomorrow.

When evolving the Web services, the engineers (producers of the Web services) will have to consider the multi-customer mode. Web services can be used by many different customers at the same time regardless of whether they are within the same organization or not. This will strongly affect their requirements gathering process.

A large number of concurrent users together with continuously emerging new business demands can result in a substantial number of parallel change requests for particular services. The total number of requests will be much higher than in traditional architectures. These requests may even contradict one another.

There will also be a need for a more global change management process and rules for co-ordinating its subprocesses run by different organisations. Web services may be composed of other Web services, offered by various vendors. When changes will have to be made, all those vendors will have to be involved.

3.6.2. Problem management. Web service systems may be distributed across many external parties. Hence, the maintenance engineers may not get an overview of the system structure necessary for identifying and analysing the reported problems. The structure will be hidden behind the service supply networks. The visibility of the architecture, the structure of the services and the implementation details will be limited to the services within the company managing the problem. At its most, the engineers will be able to identify the Web services leading to the problem. To be able to analyse and resolve the problem however, they will have to get support from the organisations providing the Web services.

3.4.3. Emergency Management. Web services may support business critical systems. The organisations will encounter many serious emergency problems. In traditional systems, the information about the emergency problems is often scarce, their causes are unclear and the actions to be taken are not always certain. Therefore, to effectively manage them, the organisations have created separate emergency processes and Task Force Groups (emergency owners) for attending to emergency problems. An example of such a process is provided in [12]. Regarding the emergency problems in Web Service systems, it is not clear what an emergency process should look line and who should be the emergency owner, when each organisation has its own Task Force Group. In [12], we illustrate how SAS has solved this dilemma.

3.6.4. Release Management. Web services affect the release management process. First, the scope of the release changes may cover small releases encompassing only one or several low-level Web services, or complex high-level ones, or even an entire Web service system. Second, fast changing business environments driving many varying change requests will lead to a large number of frequent and many times parallel releases.

To effectively manage it, the organisations will have to create a well thought-out disciplined end-to-end release management process involving all parties concerned. More effort will be put on communicating on and co-ordinating the releases, and on agreeing on their contents. In some cases, there will be a need for a global Release Advisory Board to make major release decisions.

To be able to manage the dynamic “on the fly” nature of the business processes, the organisations will have to change the nature of their release management processes, from fully planned to semi-planned. An example of such a process may be found in [13].

4. Our Framework

Using the characteristics unique for evolving and maintaining Web service systems, we have created a framework consisting of a set of suggestions for changes to the organisations, roles, and processes. They are presented in Sections 4.1 – 4.3, respectively. Most of the framework’s constituents are not new. However, the introduction of SOA and Web Services has raised the issue of their importance.

4.1. Organisational Changes

To manage Web services effectively and efficiently, we propose two organisational changes: (1) creation of totally new organisations, the so-called Service Centres, and (2) internal reorganisations within the existing software organisations. We tackle the high distribution of the Web service systems by defining primary Web service contractors.

4.1.1. Service Centres. As Harry Sneed mentioned during the IEEE panel [8], Web services introduce a concept of totally new organisations, the so-called Service
The level of trust and type of cooperation. The first category contractors. The main difference between them lies in the level ones) and not with a myriad of many different ones. Only a limited number of contractors (providing the high-level Web service management, and thereby to an improvement of evolution and maintenance processes.

This is because the organisations will have to deal with a large number of contractors and services are involved. To effectively manage the evolution and maintenance of Web services to the primary contractors (see Figure 1). To effectively manage the evolution and maintenance of Web services within a software organisation, we propose introducing a dedicated Web service team. As depicted in Figure 2, this team is responsible for managing both internal and external Web services. Our proposal concerns all the support line levels, Support Line 1-3.

4.1.2. Reorganisations within Software Organisations. To effectively manage the evolution and maintenance of Web services within a software organisation, we propose introducing a dedicated Web service team. As depicted in Figure 2, this team is responsible for managing both internal and external Web services. Our proposal concerns all the support line levels, Support Line 1-3.

4.1.3. Primary Contractors. The notion of service supply chains implies that Web service systems are highly distributed across many different parties. To effectively manage this high distribution, software organisations should mainly focus on the management of their internal Web services and outsource the external high-level services to the primary contractors (see Figure 1).

Limiting the co-operation to the primary contractors is crucial because the management of all the contractors and services in the service supply chains can be extremely expensive. It is not efficient and often not feasible, when a large number of contractors and services are involved.

Outsourcing the responsibility to the primary contractors will contribute to the decreased complexity of the Web service management, and thereby to an improvement of evolution and maintenance processes. This is because the organisations will have to deal with only a limited number of contractors (providing the high-level ones) and not with a myriad of many different ones.

We distinguish between two categories of primary contractors. The main difference between them lies in the level of trust and type of cooperation. The first category applies to the cooperating business partners. This category is derived from the business-to-business (B2B) domain, and therefore is strongly focused on the business integration and close collaboration between the contractors and business organizations.

The second category applies to the general providers offering Web services to other parties. This class of contractors is derived from the Web service business model presented in Section 3.2. In the case of general providers, the cooperation is confined mainly to the sell-buy contracts. There is no tight business integration or mutual collaboration like in the previous case.

4.2. Role Change

We have identified three groups of roles necessary for evolving, maintaining and supporting them. They are:

- **Role Group 1**: Roles responsible for creating, evolving and maintaining Web services based on the traditional systems, i.e., legacy systems. These roles must possess deep knowledge of both the traditional and the Web service systems and technologies.
- **Role Group 2**: Roles responsible for creating, evolving and maintaining only Web services and business processes. These roles must possess deep knowledge of the Web service systems and technologies.
- **Role Group 3**: Roles responsible for supporting the Web service systems at the front-end support level. These roles do not need to possess deep technological knowledge. However, they must be acquainted with the business processes and their underlying structure, so that they can support the customers in their daily operation.

Our suggestion for the individual roles is based on the new approach to understanding of the Web service systems. Due to hidden architectural and implementation details, maintenance engineers have to change their focus from understanding the implementation of the software system to understanding the purpose of the individual Web services and their role and cooperation with other services within a combined business process [3, 17, 18].

The new approach to software understanding is derived directly from the SOA concept, where the main focus is put on the business processes, the aggregations of services and business processes into end-to-end business process flows. The business processes perform particular business operational tasks and together with other business processes they constitute a high-level business process. Some of their constituents, such as Web services or lower-level business processes can be part of more than one business process.

From the architectural perspective, the complexity of the business process architecture can drastically vary from small business processes implemented with only a few Web services to the extremely compound ones, containing a large number of Web services, business processes or
part of Figure 3, we suggest the following roles for the daily build research [9]. As illustrated at the bottom part of Figure 3, we suggest the following roles for Role Groups 1 and 2: Business Process Architecture Administrator, Business Process Architecture Manager, Business Process Manager, Business Process Analysts, and Service Manager. These roles constitute an entire Business Process Maintenance Team to be operational at Support Line 3. Regarding, Support Line 1, the so-called help desk, we do not suggest any specific role. The personnel possess shallow and general knowledge of the supported software systems. They do however support all types of systems at this level, the traditional and Web service ones.

Regarding the second support line level, the personnel possess much deeper system knowledge. Their role is to confirm the reported problem in the Web services, and escalate it to the back-end support. Therefore, we dedicate a role responsible for this task, and call it Business Process Support Engineer.

Our suggestion for roles should be understood as a designation of the responsibilities necessary for managing Web service systems. They do not necessarily need to match individual persons. In some maintenance organizations, especially the smaller ones, all of them might be conducted by one and the same individual. In large organisations, on the other hand, one role might be executed by several individuals. Below, we describe the Process Maintenance Team and its constituent roles.

421. Business Process Maintenance Teams. To maximise productivity and coherence with the SOA concepts, we recommend that teams be assigned to business processes. We call them Business Process Maintenance Teams. As shown in Figure 3, one team is responsible for one or several business processes, and consequently for the Web services belonging to these processes and their interdependencies. The number of the assigned processes to one team depends on the process scale, complexity, interdependencies and the like.

The designation of business process teams facilitates the evolution and maintenance of Web services belonging to a specific business process; the services get simultaneously developed, tested, and integrated by one team, and the amount of co-ordination and planning amongst the teams is substantially decreased. Most of the co-ordination and planning is handled within the team.

Still however, some co-ordination and planning amongst the teams will be required. Due to the opportunities to reuse Web services in more than one business process, the teams should trace and manage all the interdependencies with other processes. It is crucial because changes within one business process can affect the other teams’ processes. In cases when a particular component is used by many processes, we suggest assigning it to a team, which manages (or is likely to manage) the largest number of changes in this component.

4.2.2. Service Manager. The Service Manager owns a set of particular Web services belonging to some business process. Her expertise lies on the implementation level (for the internal Web services), and on the interface level (for the external Web services). Her major responsibilities are to conduct all kinds of changes to the Web service units, such as performing enhancive and corrective changes, identifying their impact, unit testing them, and the like. The changes are requested by the Business Process Analyst or Business Process Manager.

4.2.3. Business Process Analyst. The Business Process Analyst is responsible for the business processes assigned to her team. This role closely cooperates with both Business Process Manager and Service Manager. Her responsibilities are (1) to provide necessary process architecture knowledge to the team members, (2) perform analysis of the suggested changes, assign them to the relevant Service Managers, and check that the changes have been correctly attended to, (3) take part in solution design, (4) assist the Business Process Manager in making important decisions, and (5) keep track of all interdependencies with other team’s business processes.

4.2.4. Business Process Manager. The Business Process Manager is responsible for the management of one or several teams. She possesses knowledge of all the processes managed by her team(s), and hence, she is responsible for delivering business processes and assuring their quality. She assists the Business Process
Architecture Manager and other roles in designing the overall business process. She makes various kinds of important decisions on changes in the process architecture, changes to its constituent services, and other issues.

4.2.5. Business Process Architecture Manager. The Business Process Architecture Manager has high expertise of the overall business process architecture, its constituent processes and their relationships. She is supported by the Business Process Architecture Administrator who continuously provides her with information on the status of the overall business process.

The Business Process Architecture Manager is responsible for the management of the overall business process, i.e., (1) for designing its architecture together with Business Process Architecture Administrator and Business Process Managers, and (2) for deciding on major architectural changes to the overall process.

4.2.6. Business Process Architecture Administrator. The Business Process Architecture Administrator carries out most of the administrative activities. In traditional software systems, this role may be automated. Due to the high level of Web service reuse and change volatility, however, we believe that this role is necessary, at least at the initial technology implementation phase. Her task is to keep up-to-date information about the overall business status and provide it to all the roles concerned.

The Business Process Architecture Administrator assists all the roles in various kinds of tasks. For instance, she is the focal point of contact for the Business Process Support Personnel at Support Line 2. She accepts problem reports from Support Line 2, analyses them, and assigns them to the appropriate business process teams. Together with Business Process Managers, she assists the Business Process Architecture Manager with all kinds of decision making by mainly providing feedback on the status of the business processes.

4.3. Process Changes

Introduction of the Web service software systems is going to affect most of the evolution and maintenance processes. This is due to the fact that these processes may be spanned across several parties. For this reason, all their constituent process models will have to be scrutinised and adapted to the new situation. Right now, the lead author of this paper is scrutinising some of the processes at SAS in order to find issues specific for managing the Web service systems [12, 13]. This type of work is very resource demanding and it can take time before one can make appropriate suggestions for individual process changes. Hence, for the time being, we may only suggest two major processes to be covered in our framework.

These are processes for managing Service Level Agreements and Configuration Management. These two processes constitute a baseline for optimising all the other evolution and maintenance processes and for adapting them to the new Web service application mode.

4.3.1. Service Level Agreements. A Service Level Agreement (SLA) is a written agreement between collaborating organisations [16]. It specifies what a service consumer has the right to expect from a service provider. It is a document identifying the key service targets and responsibilities of the parties involved. Its content is continuously revised by an SLA process, a process to be commonly run by the collaborating organisations. During this process, the organisations record the efficiency of their own operation and of their collaborating organisations in order to ensure that services are delivered according to the agreements made in the SLA document and in order to provide feedback for adjusting and renegotiating their responsibilities to stay in line with possibly changing needs.

In the context of Web services, we can distinguish between two types of SLAs: electronic and static. The electronic SLAs are used for automated negotiations between parties (i.e. while discovering and buying Web services) performed at runtime by software. They are intensively researched on today. One of the examples can be found in [7]. We do not consider them in our framework. We are more interested in the static SLAs to be managed on the organisational level and to be utilised between tightly co-operating companies.

The domain of the SLAs is a strongly neglected subject within software engineering [10]. Presently, there are no general SLA process models. The introduction of Web services has however raised the issue of their importance. It is caused by the fact that the evolution and maintenance of Web services is beyond the scope of one support organization and it is based on a tight and extensive collaboration with the external parties. The already-mentioned service supply chains and business model result in a set of issues and questions, which should be explicitly covered in an SLA document. The quality of the evolution and maintenance processes strongly depends on the SLA documents and the processes managing them.

4.3.2. SLA Documents. The content and the initial targets in an SLA document must be first agreed upon by the collaborating organisations. The content varies depending upon the type of SLA. However, there are common elements presented in [10]. We suggest reconsidering these elements to assure that they cover specific aspects of the Web services. Furthermore, we suggest adding new elements. They are:

- Commonly defined and performed evolution and maintenance processes: The evolution and maintenance
processes affected by the introduction of Web services should be clearly defined in terms of roles, activities, timeframes, synchronization, information/data exchange, common tools, and the like. It is crucial to agree upon how to commonly perform these processes.

- **Maintenance responsibility**: The SLA document should explicitly state who and when is responsible for the activities within the common processes.
- **Deliverables**: The SLA should specify the documents to be provided with the Web services. These mainly include documentation, and interfaces.
- **Quality factors**: The SLA document should identify quality attributes to be followed when developing, evolving and maintaining Web services, and criteria for measuring and controlling them. It should also identify technical quality factors, such as network performance, uptime availability, security, and the like.
- **Web service criticality**: It is important to identify high-priority Web Services and define how they should be managed. This assures the quality and continuity of the core business processes.

### 4.3.3. SLA Process

Defining SLA documents is however not enough. To improve their long-term collaboration, the organisations should commonly define an SLA process during which they record the efficiency of their own operation, of their contractors, and take appropriate measures.

An important prerequisite for a successful SLA process is a mature evolution and maintenance process, which in turn, would provide sufficient process visibility for recording historical evidence of the process performance. As depicted in Figure 4, the organisations should also define contractor profiles collecting information about the contractors and information about the fulfilment of the responsibilities as defined in the SLA documents and recorded in the historical process records. All this should help the organisations ensure that services are delivered according to the agreements in the SLA documents and provide feedback for adjusting and renegotiating responsibilities to stay in line with possibly changing needs. Keeping detailed records of the evolution and maintenance processes, their process steps and experience, the organisations have sufficient data enabling them to evaluate their collaborating organisations, their behaviour, and fulfillment of the assigned responsibilities. This, in turn, should help them state conditions for continued co-operation.

### 4.3.4. Configuration Management

Web service systems can be seen as a network of components with many mutual relationships. The components and their interdependencies may change much faster than their traditional components. There may also be many versions and variants of particular services at a time. For this reason, one must have accurate configuration information, and a process for managing it.

The system infrastructure should be broken down and its constituents should be uniquely identified to make possible the effective control, recording, reporting, and versioning of WS components to the level that the business requires [6].

As depicted in Figure 5, we propose organizing the configuration structure around business processes. Business processes are composed of a collection of Web services and therefore the basic configuration items (CIs) are Web services. The configuration management should provide data about all the Web services implementing particular business processes. Information regarding all their relationships with other services and processes, versions, and variants should be identified as well.

The essential issue concerning the configuration management is the level of configuration control and therefore the granularity of configuration components. Web services can significantly differ in their granularity. Particular Web services can be individual small components or the whole complex applications containing hundreds of services. Choosing the appropriate configuration component level is a matter of achieving a right balance between information availability, the right level of control, and the resources and effort required to support it. It can vary between different organizations, system architectures, and the like. There is a recognized dependency stating that changes to the software system should be made at the level recorded in the configuration repository [6]. Hence, the configuration level for the Web service architectures should be based around Web services providing distinguishable and self-contained services, i.e., looking from the business process modelling perspective. The granularity should be adjusted and consistent with the business process modelling level.
5. Feedback from Organisations Studied

We have interviewed ten companies managing Web services. However, just as many other companies in the world, they are at the initial implementation phase of the Web service technology. Right now, they use the traditional methodologies for developing, evolving, and maintaining it. This has helped them gain some experience and identify needs and thereby get some ideas about the changes to be conducted within evolution and maintenance. These ideas provide an important feedback for evaluating the credibility of our model. At this early technological phase, we cannot do otherwise. The interview results are presented below. When presenting them, we follow the order of our questionnaire in Table 1.

5.1 Progress of the organisations studied

All organisations studied manage Web services. It was difficult for the organisations to state the amount and size of their Web services. It varied from 2 to 100. Many of the Web services could consist of many small detailed ones. Eight out of the ten companies studied used external Web services, provided by their contractors. The ninth organisation is going to introduce external Web service in the near future. The tenth organisation is a typical service provider. It uses external services, integrates them into standard packages, and delivers them to their customers.

5.2 Web Service Maintenance Teams

Only one out of the ten organisations studied has a separate team for managing Web services. Seven organisations said that in the future they will have to create separate teams for managing Web services. Two of these seven organisations have experienced the need for creating separate Web service teams. Right now, their Web services are managed by the teams responsible for the traditional systems. They have however noticed that it was not an optimal solution. The substantial amount of Web services and their complexity has lead to a deteriorated quality and lack of control. Three out of these seven organisations are of the opinion that it is necessary to create new teams. Finally, two remaining organisations (2 out of 10) oppose the creation of separate WS teams.

5.3 Primary Contractors and Subcontractors

Nine out of the ten organisations studied wishes to have only one contractor (a primary contractor) for delivering external Web services coming from various sub-contractors. The tenth organisation was a Web service provider. Hence, this issue was not relevant for them.

The organisations studied prefer to have one contractor providing the final functionality. Two of the organisations however, mentioned that one should strive for having a primary contractor. However, in some cases, such as for instance emergency situations, a primary contractor can become a bottleneck. Hence, one should make a careful judgement whether to have a contact with some selected sub-contractors only in some specific situations.

5.4 Web Service Roles

None of the organisations studied has created specific roles for evolving and maintaining Web services. The design of their team structures is not process but product oriented. When studying the roles as suggested in our framework, only one out of the ten organisations could not see any point in creating process-oriented roles. The remaining nine organisations were of the opinion that the framework’s team design was very innovative, attractive, expensive, however, necessary. The organisations also claimed that they were slowly adopting this team structure.

5.5 Service Level Agreements

Eight out of the ten organisations studied have defined an SLA process, a process for managing a long-term collaboration with their partners. The remaining organisations have not done this; however, they will define such a process in the nearest future.

Seven out of ten organisations include Web services in their SLAs. The interviewees from these organisations pointed out that the content suggested by our framework was highly relevant, and that most of it was already covered in their SLAs. Especially important was to distinguish between critical and non-critical Web services and agree on how they should be attended to by.

5.6 Configuration Management

None of the organisations studied has a configuration process explicitly defined for managing Web services. They do have however a configuration management process for handling traditional systems, and all Web services are configured together with those systems.

When studying our suggestion, the interviewees were of the opinion that it was the right way of organising Web services, and they would strive to implement this solution in the future. Only one organisation, the organisation providing standard Web service solutions to their customers, was of the opinion that the framework’s suggestion was optimal for the organisations building business processes, but not for them.
6. Conclusions and Future Work

Web services systems impose additional complexity upon software maintenance and evolution processes. To handle this complexity, we have proposed a general framework for evolving and maintaining Web service systems. Our framework includes organization, role, and process changes to accommodate Web service product and use characteristics.

Our framework was presented to ten organisations in Poland and Sweden. The majority of the companies expressed their opinion that the framework’s suggestions for changes were sound, innovative and necessary for optimising the Web service evolution and maintenance process. However, the organisations claimed that they were expensive to implement. They force the organisations to transfer from product to process oriented mode.

On the basis of the opinion of our interviewees, we conclude that our framework is worth trying. When doing it, however, one should not forget to make a cost/benefit analysis in order to evaluate its effectiveness. We are cordially inviting the software engineering community to experiment with it in an industrial setting.

Acknowledgement

We thank Vinnova for their financial support of the SERVIAM project. We also thank all the individuals who have helped us with this study. They are the Peter Söderström, Björn Fagerstedt and Magnus Clarving from SAS, Oddgeir Vestad and Magnus Larsson from SEB, Örjan Carlsson from CSN, Christer Sandin from AMF Pension, Lars-Ake Hedbom and Lennart Persson from Volvo IT, Per Johansson och Kenneth Vikström from Personec AB, Jozef Ludwicki, Janusz Kozlowski and Robert Wilinski from Prokom Software SA, and Piotr Dukorno and Anita Komorowska from ATENA.

References


