Key Practices for SOA Adoption

DUŠKO VUKMANOVIĆ¹, DAMIR KALPIĆ²
¹Oracle Hrvatska d.o.o., Budmanijeva 1/IV, 10000 Zagreb, Croatia
²Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, Croatia
dusko.vukmanovic@ieee.org     damir.kalpic@fer.hr

Abstract: - Service-oriented architecture (SOA) has become a widespread architecture and integration means in companies, due to increased use and maturing of standards, technologies, tools and platforms. Platforms are certainly more mature than some other non-technical aspects of SOA. Adoption of SOA principles in system design results in certain benefits. In order to achieve balance between long-term goals and short-term business needs, companies have to establish a proper organizational and operational mode from the beginning of SOA initiatives. In general, roadmap is an accepted means of providing the plan to reach certain objectives. It links together the application, technical and organizational challenges, technology solutions, and it helps in setting priorities to achieve the goals. The aim of this paper is to identify key practices for a successful start of the transformation to service-oriented architecture.

Key-Words: - SOA, service-oriented architecture, roadmap, key practices, SOA adoption

1 Introduction

Service-oriented architecture (SOA) has become a widespread architecture and integration means in companies, due to increased use and maturing of standards, technologies, tools and platforms. Yet, SOA disappeared from the list of emerging technologies in the latest report from the middle of last year [1]. This could indicate that the benefits of SOA have been proven as well as accepted, that SOA adoption is at a high stage where 20 – 30 percent of the target audience has adopted or is adopting it and that the possibility of failure is low. Clearly, this refers to the products with which we build a service-oriented architecture. However, SOA has another, non-technical side that refers to adoption of SOA principles in system design and in obtaining of the expected benefits in return.

To produce a plan for an effective SOA program in a heterogeneous environment, a company must objectively assess strengths and weaknesses, establish strategic direction, define a practical way, mitigate risks, rationalize portfolio, manage requirements, justify the compromises and consistently examine how the execution progresses. In general, roadmap is an accepted means of providing a plan to reach certain objectives. It is developed by assessing the current situation, defining the desired future state and finally by performing a gap analysis to define steps to achieve this condition, based on the priorities and the future vision. Previous works [2], [3], [4] describe various characteristics and practices of the SOA roadmap, but none of these descriptions is sufficient for practical application. There has been little research on the key factors conducive to successful SOA adoption [5], [6], [7]. High-level and all-encompassing descriptions of the key factors make it difficult for practitioners to manage them easily. The aim of this study is to identify key practices applicable in practice for a successful SOA implementation. Similarly to the analysis of the key success factors [8], we wish to identify the minimum number of steps, which, if applied properly, should bring success.

The remainder of this paper is organized as follows: Section 2 describes the proposed reference SOA maturity model; Section 3 examines the research steps; Section 4 identifies key practices to be focused on in the initial steps and it provides their short description. Lastly, the conclusion points to the current issues and indicates directions for further research.

2 SOA Maturity Model (SOAMM)

All the roadmap developments follow the same four steps: Where are we now? Where do we want to be? How wide is the gap? How can we accomplish desired objectives (the development of actions for achieving the goal)?

The backbone of a SOA roadmap development process is the maturity model. Everything is measured against the maturity model and the activities to be undertaken in order to achieve a certain maturity level to emerge from it. In the
software industry, maturity is often related to the capability maturity model (CMM) and to CMM successor, the capability maturity model integration (CMMI) [9]. CMMI is a method for assessing and measuring the software development maturity and integration processes within an organization. The SOA maturity model reflects this understanding and it measures the SOA capability according to defined maturity levels. Maturity models are useful to assess the current position and to create a vision for the future.

SOA maturity models cannot provide any concrete roadmap because they focus on a concise description of the most important information about the architecture and not on the complex interaction of multiple factors [10]. Precisely these factors, namely, practices, are what we need to identify.

SOA maturity models are confusing, since they use different vocabulary, level rankings are different, and they group capabilities into different domains. Therefore, we define a reference maturity model to which various maturity models will be compared. In this way, we become able to analyze the capabilities that belong to the first level of maturity, since we wish to identify those practices to be overcome, in order to achieve the opportunity to reach a higher level of maturity.

The suggested maturity levels, shown in Figure 1, are based on IT and enterprise related motivation [11] and the three levels model of SOA maturity [12].

![Reference Maturity Model](image)

Figure 1: Reference Maturity Model.

At the stability level, functionalities are extracted from monolithic applications into services, and we use the service-oriented architecture mainly for integration needs. Essential characteristic of this step is that in most cases, but not necessarily, for service integration an orchestration module together with the enterprise service bus (ESB) are used.

Flexibility means that existing services are used both to build new applications and to develop the rest of required functionality. For a static composite application, developers write a new code that can be connected to existing services [13]. Dynamic composite applications can be built with a BPEL module.

Sense and respond includes modelling, monitoring, measuring and optimizing the business processes performance. There are plenty of tools for BPM implementation. Such tools are usually associated with existing applications in a proprietary manner; hence, we can end up with a strong coupling problem. In order to avoid such situation, it is strongly recommended to perform the SOA adoption first and to build BPM on top afterwards. BPM is focused on optimizing the business by observing the business processes. On the other hand, SOA is focused on the organization's IT infrastructure. However, SOA transformation alone does not provide immediate business benefit, except for IT operations.

3 Research Steps

According to the classification of the SOA research area, this paper concentrates on the SOA research question “How to design, implement, and manage SOAs?”. This type of theory is classified as the Theory for Design and Action [14].

In the first step of the research, we compare maturity models to determine which levels are compatible to the stability level. The literature review in this paper focuses primarily on SOAMMs as the most commonly used and available maturity models. In addition, their capabilities are described adequately and in detail. The SOAMMs included in this analysis are the following: HP SOAMM, IBM Service Integration Maturity Model (SIMM), Open Group Services Integration Maturity Model (OSIMM), IT Service Capability Maturity Model, Oracle SOAMM, New SOAMM (Sonic, Amberpoint, Syntinet, BearingPoint) and Microsoft SOAMM [15]–[21]. A comparative analysis approach [22] is used to transfer the analyzed SOAMMs to the proposed reference maturity model. Table 1 presents the relation between a specific SOAMM and the reference model. IBM SIMM [16] has been submitted to and adopted by The Open Group as foundation for OSIMM [17]. Therefore, they are placed in the same column in Table 1, since they share the same level of maturity.

In the second step of this research, we analyze capabilities from the maturity levels of the analyzed SOAMMs that correspond to the stability level. In formulating common practices in SOA implementation, we used a relational contents analysis technique [23] to select core practices.
Content analysis is defined [24] as “a research methodology to induce valid reasoning based on data and context”. We conducted a domain-independent analysis of capabilities, since different SOAMMs have differently defined capability domains. The most difficult part was to identify practices concealed behind the description capability, and to find out at which knowledge or application level one can expect them. In addition, the same practice was described in several capability domains and it was often applied at a higher maturity level in the broad enterprise segment. For example, in information domain one can find the description of capability [19] as „Business units are using relevant information entities”. At a higher maturity level, the description is “Key information entities have been identified at a division level“. The same capability at a specific level in [17] is described as „Initial data vocabularies are beginning to emerge“ and after that at a higher level as „Business data vocabularies have emerged but are system specific“ and „Formal business models have emerged“. These descriptions lead us to conclude that this hidden practice implies the standard definition of business data objects, which provides a basis for business object services.

4 Key Practices

A list of practices that one should focus on during individual steps of the SOA journey can be very long. The idea of key practices is to focus on the most important activities for SOA to succeed, as well as on one of the principles on which the implementation roadmap is based. Key practices derived from the analysis of maturity capabilities at the stability level of maturity are: service layers, business object services, integration centre, error handling framework, security, versioning and basic governance.

This approach to SOA transformation mitigates risks and costs by identifying the key activities that should be focused on during the initial steps, together with the reasons why they are essential. Companies that are starting the SOA transformation will choose what suits their business needs and adapt to a consistent approach of the enterprise architecture, if necessary. Descriptions for the key activities identified in this paper are offered to varying degrees. They depend on how general the activity is and how specific it can be to a particular organization and the selected SOA platform.

4.1 Service layers

Layers of service are part of the SOA reference architecture (SOA RA). In this paper, a division of services into three layers is proposed, as shown in Figure 2: utility, business and the orchestration layer. The division is based on [25] and [26], with additional elements.

Utility service layer is a technologically oriented service layer, it is re-usable, it contains present-shared resources and capabilities in the company, but it does not include business logic.

Designing of the business services layer leads to the creation of two types of business services: task service and business object service. Business object services include the logic associated with essential business entities, such as a purchase order or a customer. Task service includes business logic specific to a task or the parent business process step, focusing on a static business function. When developing a business service with business logic,
we often insert the logic that belongs to the utility service and thus it may result in a hybrid service.

Orchestration service layer indicates a business process or task. These services are also task-oriented, only representing a business process defined in orchestration platform, which is exposed as a web service. The services can encompass a complete business process or an independently defined sub-process.

4.2 Business object services
As it was explained in the preceding paragraph, the initial efforts in developing a service should be organized as a business object services development. In order to support the development of these services, SOA data model must be defined. It consists of a standard definition for business data objects and a re-usable information component that represents the business object, such as purchase order, invoice, or a customer. These canonical definitions of the object gather different applications using a common language and drastically reduce the number of transformations needed for integration between different applications. Their development will build a layer of data services in one’s reference architecture. This approach will ensure the development of the interface first, which forces the organization to take a strategic approach to enterprise architecture, rather than to develop a tactical point-to-point integration.

4.3 Integration centre
We need someone to lead the SOA effort, where we centralize SOA operations, someone to support us until we reach a greater maturity, someone who is actually responsible for looking forward. In order to achieve the strategy, organizations must set up proper guidance. Depending on whether the activity can become a standard practice or whether there is a continuing need for leadership and management, guidance may or may not be permanent.

At this stage, there are two levels of required effort. On one level, companies need to establish the SOA capabilities continually. On the other level, assistance needs to be provided for each project and the SOA should result with total control ensured. The team engaged will play a significant role in support to the execution of the first process, as well as in creating a support team.

4.4 Error handling framework
Error handling framework provides users with a consistent way to manage errors through technology in the integration layer. It provides an opportunity for routing an error back to the application (process) that it has caused and to its user. In this way, exceptions can be resolved quickly with less downtime and IT can meet stringent service level agreements. Operation and maintenance of distributed composite applications requires a robust management for the errors and problems solving system. Together with the error-handling framework, it is necessary to form a team for support of the SOA systems. For the start, it can be an integration team.

4.5 Security
If the initial SOA efforts are related to integration or processes within the enterprise, security requirements are not likely to be high. In addition, services are most often accessed via portals or through any other interface where authentication and authorization have already been completed. However, when we expose services to the outside actions or we increase the number of services, so that no longer we can believe that they would always be invoked through an interface performing authentication and authorization, we must use the security standards for web services, depending on the level of security that we want to achieve. For this purpose, we should use security modules that are part of the SOA platform.

Security dimension of web services [27] includes secure messaging, resource protection, contract
negotiation, trust management and security properties. Security decisions must always be made with understanding of the threats faced by the system that we want to protect.

4.6 Versioning
Autonomy is one of the fundamental concepts of orientation to services, which requires that services can be deployed, updated and maintained independently of each other, as well as solutions that use them. An autonomous item has the freedom to control decisions without requiring the consent or adjudication from outside. With increased service autonomy, greater reliability and predictability are achieved. It is precisely through versioning that we want to ensure the necessary service autonomy.

In order to implement versioning, the following must be defined: versioning unit (whether we version the service and related artefacts together or independently), service changes (for what changes the new version occurs) and the way of deploying new version and access. If we follow the current recommendations in the previous paragraph, we need a mode for scheme versioning, because the business objects are described with XML schemas, services and applications participating in the integration.

4.7 Basic governance
Governance is more important in an SOA environment than in the current IT environments. Shared life cycle management is different from the traditional applications life cycle and therefore the burden of the success of SOA architecture is shifted to governance.

At the beginning of the introduction of SOA architecture, we may have a situation where we have set a very demanding delivery dates for key functionality. There would be not enough time left for governance planning and implementation. One should be familiar with SOA governance and understand the risks of insufficient management. Although one does expect some problems in the course of SOA implementation, business will not wait until next year to set up the required processes. Therefore, we need a basic governance to survive till the next SOA project. SOA governance refers to all activities that are not immediately required for the services development. Nevertheless, it increases the overall SOA quality and enables control in complex environments. SOA governance is a combination of people, policies and processes that organizations use to achieve the desired behaviour.

Basic SOA governance suggested for a quick start includes elements already been defined in the preceding paragraphs, as key practices indispensable for the success of SOA initiatives.

5 Conclusion and Future Work
Enterprises want to modernize their applications using SOA and wonder how to successfully adopt SOA. We should not forget that the resources, people and money are limited and everyone is looking for a fast solution. Hence, there is an urge to begin the development in line with SOA principles almost immediately, yet without making a mistake that could cost dearly in the future.

The paper highlights the key practices that should be focused on, so that one can start immediately with the adoption of SOA architecture. The objective of this work was to minimize the number of these practices while trying not to omit any important practice, what might later cause problems. There are many tools to be mastered and activities to be completed during SOA adoption so that it is very likely that the implementer may not have enough “power” to do everything in accordance with best practices, particularly regarding the activities that are less exact, such as services design. The focus is on the gradual implementation of SOA concepts to existing information technology in order to obtain short-term business benefits and ensure the success of SOA in the future. As we progress through maturity levels, we will introduce new practices and improve the existing practices.

Future study could take the form of a case study analysis in order to confirm or disprove these practices.

References:


