SOA Implementation
Strategy
# Table of Contents

1 Introduction .......................................................................................................................... 2
2 Stage 1: Establish the SOA Foundation .................................................................................. 4
3 Stage 2: Define Your SOA Strategy .......................................................................................... 6
4 Stage 3: Apply, Measure, Iterate ........................................................................................... 12
5 Appendix A – Typical SOA benefits ...................................................................................... 13
6 About SOA Software .............................................................................................................. 14
1 Introduction

This document describes a pragmatic approach to implementing SOA in a way that helps ensure that companies attain the benefits of SOA.

Too often companies implement SOA as a technology solution; while they do increase complexity with many loose-coupled services, that approach fails to deliver the benefits of SOA. SOA can only be successful if it is well defined and well managed. The basis for this is to define the objectives that you expect from your SOA initiative.

Don't think that you can derive these objectives from generic SOA advice. Instead of drawing up SOA objectives in a vacuum based on the best guesses of the people writing them, you should derive the objectives from both IT and business strategies.

To understand why this is important, let's look at a couple of simplified examples:

Company ABC’s current strategy is to grow aggressively, including through mergers and acquisitions. The IT strategy should be to support this growth, for example by allowing for several competing applications to run for a while until their functionality is merged. In turn, one of the key objectives of ABC’s SOA initiative should be enabling business agility through quick on-boarding of acquired companies.

In contrast, company DEF’s business strategy is to re-focus to concentrate on the core business and reduce costs. In this case the IT strategy should focus on being optimized for the core business, for example buying best-of-breed industry-specific applications. For company DEF, business agility would probably not be among the core SOA benefits, but reducing costs and a closer link between IT and the business might both be high on the list.

Once a company has defined these prioritized benefits, everything in the SOA initiative should derive from them. This is not theoretical advice. Down to the lowest service design decision, everything should be traceable back to the objectives.

Continuing on the example: one of the key SOA objectives identified by ABC is the quick on-boarding of acquired companies. As a result the Reference Architecture allows for various message exchange patterns but has guidelines for using each of them. These guidelines will drive the design decisions of individual services. In this context, if a service is implemented, say using a synchronous request-reply pattern, this can be justified by the guidelines defined in the SOA reference architecture, which itself derives from the SOA objectives, themselves ultimately linked to the company's business strategy.

The SOA objectives should directly guide the 3 strategic documents driving the SOA:

- The SOA Reference Architecture
- The SOA Roadmap
- The SOA Governance Strategy

Together these documents define the SOA Strategy and allow you to implement your SOA, confident in the knowledge that you can achieve your objectives. The last thing to remember is that while you will be implementing your SOA, you should always keep an eye on reviewing and improving your strategy.
In summary, the approach described in this document focuses on establishing clear SOA objectives from the start and using them as a driver for all decisions going forward. This is the first stage: establish a foundation.

You then move to the second stage: defining your strategy through the description of the architecture, roadmap and governance for your SOA.

The final and third stage is the actual implementation and continuous review of the strategy.
Stage 1: Establish the SOA Foundation

Where do you start?

To get a good foundation for SOA Governance you need 1. a direction for the SOA based on business and IT strategy and 2. authority to apply this strategy.

The best way to combine the two is to involve top IT management in defining your SOA objectives. This approach has many advantages:

- The SOA objectives are clearly linked to the business and IT strategy
- Because they define it, your CIO and IT management team get to own the direction given to the SOA initiative. They are more likely to endorse the detailed strategy that you will define and empower you to see it through with funding and other support.
- You get the authority to apply the strategy and drive the SOA initiative.

Step 1 – Executive IT workshop

The first thing we recommend is to prepare, organize and run a strategic SOA workshop with top IT management in your company. We have seen the following high-level agenda used successfully:

1. Definition of SOA, what it is and why do we care
2. Case studies from similar companies in the same industry
3. SOA Strategy definition (a summary of this document)
4. Restate business and IT strategy – you should be this run by the CIO prior to the meeting to avoid any surprise
5. Workshop: define the prioritized objectives for the SOA

The last point is of course the main objective of the workshop. A simple method is to ask every participant to write down 3 objectives, collect all the objectives on a flipchart and work together to define the key 3 to 5 objectives in order of importance.

To help kick-start the process and sanity check the results, you may also provide a list of typical benefits companies get from SOA (see appendix A for an example).

Step 2 – Anchor the SOA Objectives

Once you have defined your objectives through a workshop or by other means, you still need to make sure that the final list is agreed and communicated.

The best way to do this, after the list of benefits and their priority has been formally agreed by IT management, is to officially "launch" the SOA initiative. How you do this will depend on your company’s culture, size and accepted means of communication.

Very large multinational corporations, for example, should restrict the communication to the teams which will have an immediate impact or be impacted by the SOA initiative. Typically that would mean stakeholders in a single department or program.
An email to all likely stakeholders (say architects and IT managers within that department) introducing the SOA initiative and its objectives has worked well for some companies. In other cases you may want to have a formal meeting or presentation to ensure that key stakeholders are on board, understand what you are trying to achieve and can ask any question they want. Companies should also publish their agreed SOA objectives on a wiki or intranet page so that people can always refer to them if they need to.
3  Stage 2: Define Your SOA Strategy

In this next stage, you define the detailed strategy for implementing your SOA. At every step you should keep in mind the objectives of the SOA. Most, if not all of the decisions should be directly linked and even explicitly justified against the SOA objectives. However, pragmatism should always be an implicit objective here and while the SOA objectives define your end goal, pragmatism should give you the path to this end goal.

Although we define this stage in 3 steps, your best approach is to run these steps in parallel or in an iterative fashion, feeding off each-other. For example, scoping the SOA initiative is part of the roadmap but should be the first activity, as it influences everything else. The roadmap should be driven in large part by the SOA reference architecture but should also consider the implementation of the governance strategy.

3.1  Step 1: Define the SOA Reference Architecture

The SOA reference architecture is, among the 3 components of the strategy, probably the easiest to define and most widely documented. Easiest because in IT we are usually comfortable with working out architectures, most documented because most commentators of SOA have an architect background. In fact, most SOA initiatives – even the most technically and least strategically focused – probably have an SOA reference architecture of one form of another.

We will therefore only include general guidelines describing what we believe should be contained in the SOA reference architecture, without explanation of how this should be arrived at.

The SOA Reference Architecture should at least include the following topics:

1. Shared Services
   a. Definition of a service
      → including separation of concerns, service consumers, etc.
   b. Service types and service taxonomies
      → We recommend distinguishing between business services (first class citizen and the subject of most of the SOA efforts) and technical services (needed to oil the SOA, but not a priority)
2. Standards, guidelines and patterns
   a. Technology standards
      → SOAP, etc.
   b. Design guidelines and patterns
      → including design patterns, service modeling, naming conventions, message exchange patterns, service versioning, etc.
3. SOA Infrastructure
   → Defines the platform for service development, mediation, governance, security, monitoring, etc.
4. Operation, Administration and Management
   → How services are deployed, run and monitored
5. Links with other initiatives
   → based on the current initiatives in the enterprise: BPM, cloud, etc.
3.2 Step 2: Define the SOA Roadmap

The SOA Roadmap defines the steps your enterprise will take in the next few years for implementing the SOA Strategy and achieving the SOA benefits.

It should not include every detailed step for the length of the roadmap as it would be very time consuming to define and in all likelihood it would be wrong the minute you start the implementation. Instead, pragmatism should guide you here: the roadmap should be very clearly defined for the first few months and more and more open the further in time it gets.

As with all the elements of your SOA strategy, the roadmap should also be a living document that you refine and update as you go along, but which helps you make sure you don't miss any important steps.

Ideally you should represent the roadmap graphically to ease communication to, and achieving buy-in from key stakeholders, including IT management.

3.2.1 Short Term

For the first few months, the roadmap should include the one or two projects that you have chosen to first implement SOA, or if SOA has already been used, where you will first apply your new SOA strategy. It should also define to what extent the governance strategy and the reference architecture will be implemented – remember to be pragmatic: focus on the easiest parts that will bring you the most benefits. You may also want to include one or two of the riskiest aspects of the strategy so that they can be tried out and de-risked.

3.2.2 Mid to Long Term

Longer term, the roadmap should specify when and in which order all the aspects of the strategy should be implemented.

An example item on the roadmap might be one element of the infrastructure. Say a service security gateway for exposing services to partners. The trigger or target is not necessarily a specific date: it could be dependent on other factors such as when a certain number of services are exposed to partners. It helps however to estimate a date for when this trigger should be reached.

It might also include when the platform or the SOA is rolled out to different departments or business units.

3.2.3 SOA Objectives

Ideally you should also include in the roadmap when you expect to reach certain target objectives of your SOA. For example if one of your objective is "reduce partner on-boarding time", you should specify when it would be reduced by say 20%, 40% and 60%, based on how the key systems that are integrated with partners are SOA-enabled.

3.3 Step 3: Define the Governance Strategy

An important part of the strategy describes how you will govern your SOA. Without governance, you will likely not follow your SOA reference architecture, your SOA roadmap will falter and ultimately you will not achieve your SOA objectives.
Most if not all IT departments are organized and funded on the basis of delivering successful projects and programs. However, SOA is based on the principle of service reuse and of collaboration between these projects and programs for the greater good of the enterprise. This conflict between a project’s immediate needs and the strategic need of the enterprise underpinned by SOA is what makes SOA governance so essential.

3.3.1 Decide on a Governance Style

The governance processes must be adapted to the organization:

- Are you in an organization where processes are formal and generally followed or where every department and project works in its own way?
- Are you implementing SOA in a large multinational in multiple business units with conflicting strategies and cultures or in a single cohesive department?
- Is your organization quick to implement new technology or usually relies on proven tools?

The answers to these questions and others will help define the ideal governance style for your SOA initiative. Some of the variations in the style include:

- Centralized vs. Federated. Can a single set of governance processes and a single governance team handle the whole SOA initiative, or do you need to be flexible to allow variations per department or business unit?
- Informal vs. Formal. Do you define guidelines rather than rules and allow exceptions or can you get away with being very strict in enforcing the rules?
- Manual vs. Automated. Can you implement your governance with a set of spreadsheets and manual checks or do you need to automate the processes and checks?

On this last question we recommend automating as much as possible to make governance effortless. We have found that when governance is mostly a manual process, practitioners see it as an overhead and try to bypass it as soon as they can.

Another aspect that is useful to look at before defining your SOA governance processes is the existence and adoption of other IT governance processes, such as IT governance (as in ITIL), architecture governance and development methodology. Examining these processes will not only give you clue as to how easy your SOA process will be to implement but will also give you hooks you can use to introduce your SOA governance processes. For example if every project goes through an architecture review on completion of the design phase, you can very easily re-use this step to do additional checks for SOA.

3.3.2 Basis of Governance Processes

A governance process does not need to be a complex workflow diagram. It is easier to think of it as a simple definition of who, what, when.

- Who: which roles or groups are involved in the process
- What: which tasks compose the process
- When: what triggers the tasks – most of them can be re-used from existing processes or methodology

You simply need to define "who, what, when" for the most common processes. The most important processes include:
• SDLC – Service production: defines how services are created.
• Service consumption: defines how services are reused.
• Exception management: defines how to handle exceptions to the rules, for example how do you recover from an exception. By defining this as a separate process it simplifies the other processes which can concentrate on the happy path.
• Portfolio management and project prioritization: defines how you identify, prioritize and allocate services that will benefit the whole organization rather than the ones implemented for project needs.

Other processes that you may want to formally define include:

• architecture and infrastructure management
• governance management

3.3.3 Who
To define the roles and groups that might be involved in the SOA governance processes, start with reviewing the existing organization, including existing governance groups such as review boards.

Then ask the question: Is any new group or new role required to better achieve the needs of the processes?

Some example new groups that enterprises have found useful to set up include: a center of excellence or competency center, a review board, a service factory. Repository librarian is a common role that may be considered.

Most of the roles and groups which will act in the processes actually exist in the enterprise today: IT management, chief architect, development teams, operation teams, etc. So it's best to start with the existing organization and only add to it if a task does not fit with any existing group or role.

3.3.4 What
To define the "what" in the processes, divide each process into a list of simple tasks. For example the service SDLC process will include such tasks as design, develop, etc.

Then draw RACI charts for each of the processes, plotting the groups and roles against the list of tasks and filling them with letters R, A, C, I. RACI stands for Responsible (executes the work), Accountable (approves), Consult (helps for the task), Inform (is informed of the result).

Some guidelines about RACI charts:

• Each role can have more than one letter for a given task, most commonly RA or AC.
• If a role has RA, it means there is no formal approval for the task.
• If there are two "A" for a task, it usually means there is a multi-level (serial or parallel) approval.
• Of course a role may have no letter (i.e. no involvement) in a task.

Below is a sample RACI chart (only the first few rows are complete) for service production.
Note that in this example the first line contains several A, not because of multiple approvals but because the domain owner, business analyst and project architect are all responsible (and accountable) for identifying service candidates.

3.3.5 When
RACI charts are a good starting point for defining processes. If not self-explanatory, more details can be added for a task, for example adding its trigger. It is also possible but not mandatory to draw workflow diagrams. When no diagram is defined, assume the tasks follow each other linearly.

3.3.6 Funding
As the goal of the SOA governance is to bridge the gap between projects and enterprise needs, some larger enterprises will want to look at defining a funding model for SOA. The funding model answers the question: who pays to modify or reuse a service.

Quiet often a service which was implemented for a project will need minor enhancements to fit another project. This would not be the case if every service would be designed to be perfectly generic but this not achievable in the real world. Enhancements may be functional (e.g. add a field in the service response) or non-functional (e.g. make the service respond quicker or support a higher load).

In most cases the enhancement is most effectively implemented by the group owning the service or the group who originally developed the service. This could be a project team or a generic team such as a service factory or service maintenance team. In all cases funding issues might arise which the funding model should answer.

Examples of funding models include:
• "tax on SOA" on all project. It's easy to implement (if not popular) and helps promote SOA (if we pay for it we might as well use it). But it can only be implemented if SOA is one of the most strategic IT initiatives.

• "pay per use" for reuse cross-departments. A more complex model to implement and which might be a counter-incentive to reuse. However, it may encourage the creation and advertising of good quality generic services.

• Central funding out of the IT budget. Very simple to implement and more popular than the tax on SOA but it might be difficult to control the budget in a fair and balanced way.

• Service factory or service maintenance team. Specific SOA teams which are paid by or embedded in the projects. This helps with the standardization efforts as all services are built by the same core team.

In all cases it is usually better to only look seriously into funding until after the enterprise is mature enough in SOA and understands the needs in details.
4  Stage 3: Apply, Measure, Iterate

4.1  Apply

Once your strategy is fully defined, the only thing remaining to do is to roll up your sleeves and actually implement it.

As you do so, don't forget to communicate successes and changes in strategy to the enterprise.

4.2  Measure

As you probably understand by now, the SOA initiative will only be successful if it meets its objectives. And the only way to find out if it is meeting its objectives is to measure it.

As part of your roadmap, you should have defined when you reach certain objectives (see section 3.2.3 above). It is important to monitor and measure these benefits in order to ensure that you are on track and if not to understand what aspect of the strategy should be reviewed.

One of the metrics you should definitely consider is the alignment to the reference architecture. You may for example check the number of exceptions to the guidelines accorded to projects. This will help determine how successful the governance is and if you are likely to achieve your objectives.

Another important aspect of measuring the success of your SOA initiative is that you can chart progress to present back to your IT executive team. This will help keep senior management on board and might even trigger healthy competition between IT departments as you present their relative progress.

4.3  Iterate

As you go through executing your roadmap, putting in place your governance and applying your reference architecture, you will find that things can be improved or refined. Every component of the strategy, roadmap, governance and architecture, should be living documents which it is absolutely essential to keep refining and modifying as the enterprise mature in its adoption of SOA.

As your SOA reaches maturity, other factors will begin to influence the SOA initiative, including change of IT strategy (often when a new CIO comes on board), adjustments in business strategy, and advances in technology. In some cases, it is worth redoing a full cycle: reviewing the SOA benefits to ensure they still align to the business and IT strategy and updating the SOA strategy accordingly. In other cases, only certain aspects of the strategy may need to be refreshed. For example moving to a platform-as-a-service SOA platform will likely mostly impact the reference architecture.
5 Appendix A – Typical SOA benefits

This list is of course not exhaustive but it does provide a good starting point.

Most generic benefits

- Business agility- although this is an often-cited benefit in SOA literature, it is not necessarily a good objective as it is too vague: try to define what it means for your business
- Close the gap between Business and IT
- Shorter time to market
- Development cost savings or IT productivity gain

Other benefits might be linked to a specific pain or strategy for the business or IT:

- Faster partner on-boarding
- Simplify on-boarding of acquired companies
- Enable IT transformation
- Help reduce the number of back-end application
- Enable multi-channel delivery
- Enable outsourcing of development
6 About SOA Software

SOA Software is a leading provider of unified governance automation products that enable organizations to successfully plan, build, and run enterprise services. The world’s largest companies including Bank of America, Verizon, and Pfizer use SOA Software solutions to transform their business. For more information, please visit http://www.soa.com.

SOA Software, Policy Manager, Portfolio Manager, Repository Manager, Service Manager, and SOLA are trademarks of SOA Software, Inc. All other product and company names herein may be trademarks and/or registered trademarks of their registered owners.