Definitive requirements for delivering successful SOA-based solutions.

Anthony Kesterton, SOA technical lead, financial services sector, United Kingdom, Rational software, IBM Software Group
Why do requirements matter?

While more organizations are now embracing service-oriented architecture (SOA), many have not yet moved beyond small-scale, technology-driven SOA projects. In this context there is a tendency to downplay software development best practices, in particular requirements management.

The fact is that successful SOA projects, whatever their scope, demand more attention to requirements than conventional software development efforts. This is because the key drivers for successful SOA solutions come from the business, and the services produced have the potential to be used across multiple solutions.

This paper describes best practices for requirements definition on SOA projects, including the key requirements artifacts to focus on, and why they are important.
In developing SOA-based solutions, there are several requirement types you need to define.

<table>
<thead>
<tr>
<th>Requirement type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business goal</td>
<td>A business goal is a requirement that must be satisfied by the business. Business goals describe the desired value of a particular measure at some future point in time and can therefore be used to plan and manage the activities of the business.</td>
</tr>
<tr>
<td>Business use case</td>
<td>A business use case is the detailed specification of a set of actions performed by a business. It yields an observable result that is typically of value to one or more business actors or other business stakeholders.</td>
</tr>
<tr>
<td>Business rule</td>
<td>A business rule is a declaration of policy or condition that must be satisfied within the business. Business rules can be captured in models, documents or both.</td>
</tr>
<tr>
<td>Business task</td>
<td>A business task is a step in a business process that identifies some action that must be performed but does not further model the details of that action. A task has a (possibly implicit) task definition that specifies its name and input and output data.</td>
</tr>
<tr>
<td>Functional requirements</td>
<td>Functional requirements present a comprehensive description of how the system will function from a user perspective. They outline a software capability that must be met or possessed by a system or a system component to satisfy a contract, standard or desired feature.</td>
</tr>
<tr>
<td>Nonfunctional requirements</td>
<td>Nonfunctional requirements dictate properties—such as performance/latency, reliability and security—and impose constraints on the project or system. They specify attributes of the system rather than how users would interact with the system.</td>
</tr>
</tbody>
</table>
Why requirements are so important on SOA projects

A perception sometimes voiced in SOA consulting engagements is, “We’re doing SOA; therefore, we don’t need requirements.” On projects like these, work starts with a proposal for potential services, and then moves quickly to service definition and implementation. To understand why this is problematic, you don’t need to look beyond the definition of SOA:

*An approach for designing and implementing distributed systems that allows a tight correlation between the business model and the IT implementation.*

The emphasis is on the deep connection between the business and IT. The key to collaborating with business stakeholders is to effectively capture their input in the form of requirements in order to align the solution with the business. The connection to business needs is what SOA is all about: SOA-based solutions need to be aligned with business goals to be relevant to the business and to deliver real, measurable value.

Another key reason to emphasize requirements management on SOA projects is risk mitigation. Because services are more “exposed,” they present greater risk to the business than typical applications. Failure of a service that all your major customers depend on has an impact way beyond the company walls. Paying attention to requirements reduces the risk that a failed solution will negatively affect customer relationships, company image and/or profits.
Yet another reason to focus on requirements for SOA projects is to facilitate reuse. Starting from better requirements and documenting a service appropriately make it much easier for other service consumers to become aware of, quickly understand the value of, and use an existing service rather than building a new one. Furthermore, because they’re built for reuse, service assets are impacted by frequent updates and changing interdependencies such that effective cross-organizational visibility of key requirements is imperative for efficiency’s sake.

Carefully defined requirements are also essential to bridging the gaps in cultures and perspectives that increase the difficulty level on outsourced and other distributed development projects where contributors are dispersed, turnover is consistent, and skills and roles are diverse.

Requirements also drive test planning on all SOA projects, and they’re a primary input into the service identification process.

Different SOA solutions need different requirements
In general, requirements for SOA solutions address the question, “What will the solution do for the business?” However, best practices for gathering and managing requirements differ somewhat depending on the scope of the SOA project.
From a requirements perspective, projects involving SOA-based solutions typically fall into one of two categories:

- Large-scale SOA-based solutions create composite applications that solve significant business problems and cross both internal and external organizational boundaries.
- Small-scale SOA-based solutions may involve creating one or several low-level services, and often are more technically driven than bigger SOA projects.

Some SOA projects are initiated by service consumers who wish to identify and reuse existing services or to drive the creation of new ones; which means many service creators will have many different requirements needs (and starting points).

**Requirements on large-scale SOA projects**

Large-scale SOA projects tend to involve the transformation of core business processes. To successfully align a business-critical solution with the business, the project must capture or reference artifacts associated with the appropriate business process and business models.

On traditional software projects, you typically separate business process/business modeling artifacts from requirements artifacts; the former serves as input to the latter but is not its focus. Many traditional projects actually ignore the business processes they are to support (at their peril). On SOA projects, however, business modeling and requirements management are closely related because every project should revolve around what the solution must do in support of the business.
As such, work products usually associated with business modeling are closely coupled to requirements on SOA projects. Developing both the business model and the requirements is vital to success. As defined in the IBM Rational® Unified Process® (IBM RUP®) solution, relevant work products can include:

- Business goals.
- Business metrics and key performance indicators (KPIs).
- Business process models.
- Business rules.
- Business use-case models.

![Diagram showing key requirement types for SOA projects and their relationships. Projects should meet the specific business goals, with all the related requirement types supporting these goals.](image)

*Figure 1: This diagram shows the key requirement types for SOA projects and their relationships. Projects should meet the specific business goals, with all the related requirement types supporting these goals.*
The business use-case models capture the key functions of the business from an external perspective, while the business process models capture the details. Because the solution must be aligned with the business, the nature of the business process and the function to be implemented significantly impact requirements.

Nonfunctional requirements also tend to be significant on large-scale SOA projects. On many SOA projects where services are exposed outside the organization or across lines of business, it is especially important to extract and define solution-wide, nonfunctional requirements such as behavior under load, failover and standards conformance. These requirements are critical because they affect multiple service consumers, who may be challenged to agree on them. For example, a service-level requirement for transaction volumes might be appropriate for one consumer, but totally inadequate for another, high-volume consumer.

At a minimum, you should gather nonfunctional requirements to cover:

- Performance (latency/throughput).
- Security measures.
- Logging/auditing.
- Monitoring.
- Configuration management.
- Failover.

Additionally, a set of operational requirements must be collected to address topics such as how to replace an existing service with a new one while a supported business process remains operational. These requirements can be defined in tools such as the IBM Rational Requirements Composer application, and they can be managed in IBM Rational RequisitePro® software.
Highlights

Business goals
According to RUP, a business goal is:

… a requirement that must be satisfied by the business. Business goals describe the desired value of a particular measure at some future point in time and can therefore be used to plan and manage the activities of the business.

Capturing business goals as requirements for SOA-based solutions is fundamental, because SOA-based solutions are meant to pull together what the business needs in order to accomplish and support its objectives. Capturing business goals enables the organization to directly drive the SOA project. And it provides IT with the context to operate at a level of abstraction while still developing services that meet business needs.

Therefore, in defining business goals as requirements, it is important to move beyond the vision level to the definition level, answering such questions as “By how much?” and “By when?” For example, “Reduce accident claim processing costs by 10 percent by December 1, 2008” is more testable and verifiable than simply “Reduce accident claim processing costs.” This ability to set quantifiable goals also supports the activities associated with prioritizing changes for multigenerational projects, and to appropriately adjust project scope to meet expectations.

Business metrics
Business metrics and KPIs, which explicitly measure performance against business objectives, are directly measured or derived from business activities. These measures are important for SOA-based solutions because they define what data is required to determine whether business goals have been met. Deriving metrics and KPIs as requirements can help analysts uncover correlations between processes that must be considered; they also have the important side effect of driving IT/business discussions around costs.
Definitive requirements for delivering successful SOA-based solutions.

Page 10

The business process provides vital input into the identification of candidate services.

Business process model

The business process or subprocess to be automated by the SOA-based solution serves as a source of input on both requirements and constraints. It embodies tasks that may be automated; defines roles that perform the tasks; and identifies the interactions among roles.

Most importantly for SOA-based solutions, the business process provides vital input into the identification of candidate services (as suggested in RUP®). But the business process is just one of many requirement sources and one of a variety of inputs into service identification. Others include:

- Existing application and data analysis.
- Goal-service modeling.
- Business use-case models.

A challenge here is that many organizations have not yet documented their business process, or its description does not match how the process actually works. When defining and designing services, it’s critical to do your due diligence—otherwise you may use valuable resources to build an SOA-based solution that supports a process that doesn’t happen, rarely happens or happens in a way that the defined service doesn’t help improve.

To understand the business process, you must capture the as-is business process model (to understand the problem) and construct the to-be business process model (as part of the solution). Not only will this help you clarify some of the functional requirements for the solution that satisfy the business vision, but it also will help you identify subprocesses that are good candidates for automation, as well as metrics, business rules and other information related to the business process.
To simplify the process of capturing requirements associated with business modeling, Rational RequisitePro includes useful templates for documents and requirement types. Traceability structures link business rules to business goals, KPIs and metrics. Capturing this information in a tool like Rational RequisitePro can make it easier to respond when business goals change, which drives corresponding changes elsewhere. Also useful on SOA projects are integrations between Rational RequisitePro and IBM WebSphere® Business Modeler software that support direct linkages between requirements and the business process model.

Business rules
The Business Rules Group defines a business rule for IT systems as:

…a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behavior of the business.

By definition, business rules are both capabilities and constraints for SOA-based solutions, and they need to be encoded in the solution. Often it makes sense to isolate business rules from other components of the solution so that frequent changes to the rules can be made without requiring major changes elsewhere.

Business rules can be captured in a variety of ways, such as Unified Modeling Language (UML) elements in a UML model, text in Rational RequisitePro or as part of the business process model (which can be captured using IBM WebSphere Business Modeler).
Business use cases

Business use cases focus on the behavior of the business from an external point of view. The external viewpoint could be that of the customer, supplier, regulator or anyone else directly interacting with or affected by the business. To describe this viewpoint, organizations should use business use-case modeling. According to RUP, a business use case:

… defines a set of business use-case instances, where each instance is a sequence of actions a business performs that yields an observable result of value to a particular business actor. A business use-case class contains all main, alternate workflows related to producing the observable result of value.

A business actor represents an external point of view; a business process is more internally focused.

A business actor represents an external point of view. Business use cases and business processes are closely correlated, but the focus is different for each. Business use cases present a concise overview of what the business is trying to achieve. This ability to provide a top-level viewpoint of the business—what it does and who it interacts with—is what makes business use cases so important to requirements definition for SOA-based solutions. ¹⁰

A business process is more internally focused because it describes how the sequence of actions flows. As such, business processes are closely aligned with business use-case realizations,¹¹ which can be conveniently held in the business analysis model in the IBM Rational Software Modeler or IBM Rational Software Architect application.¹²
Requirements on small-scale SOA projects

As mentioned earlier, small-scale SOA projects involve creating one or a few lower-level services, and are more technically driven than large-scale SOA projects. Another key difference is that small-scale projects often involve the identification and consumption of an existing, reusable service.
Service creation

From the service creation perspective, developing a single, small-scale service still entails requirements gathering, just like any other project. Requirements should comprehensively answer questions such as:

- What should the service do?
- What quality of service is required?
- What service-level agreement should be in place?
- What service policies need to be adhered to?

Service consumption

In many ways it can be harder to find an existing service than to build a new one, particularly if the service you’re searching for is local and has a tactical focus. Some of the requirements that service consumers need to identify are the same as those for service creators. But service consumers must also identify other requirements, such as:

- Who provides the service?
- What is the cost of the service?
- How long can this service be relied upon before it is retired?
- When the service is changed or upgraded, how will the processes that use the service continue to operate?

As the number of services in your organization increases, managing those services becomes an issue. You can use infrastructure tooling such as IBM Rational Asset Manager software with IBM WebSphere Service Registry and Repository software to help you store, find and reuse services and other assets. WebSphere Service Registry and Repository also helps route service calls to the appropriate services during run time, both to manage load issues and to mitigate some of the service consumers’ concerns around service upgrade and retirement.
Identifying candidate services
Since you’ve done the work to gather requirements (business use cases, business rules, etc.), you should leverage them for optimal value, such as identifying candidate services to be created or reused. Requirements can help drive, or serve as input to, service identification at the solution analysis/design level. A development process (RUP, for example) can help you sort out which services are actually required.

One way to leverage requirements for service identification is to examine the tasks and roles that are embodied in the business process with an eye toward which tasks to automate. Another way is to look at the business use-case realization, which describes in detail the activities of the business use case and who undertakes them. Interactions between the outside world and the internals of the solution are particularly useful for identifying services that have high external visibility.

Other important requirements issues
In addition to gathering and managing requirements, you should consider other requirements-related issues at the solution and service creation/consumption levels of SOA projects.

Requirements validation
As your organization begins to adopt widespread use of services, you’d be wise to establish an architecture board. This group reviews services across the enterprise and ensures that requirements are in line with the competing demands and priorities of various business units. These activities facilitate service reuse and enforce service-related decisions such as service policies.
Highlights

As you begin to use services enterprise wide, it's a good idea to establish an architecture board.

Small-scale service creation
Here, too, an architecture board with top-down responsibilities can decide whether a new service should be created or an existing service reused. Similarly, service creators should explore their requirements with potential service consumers to validate whether their service will fit consumers’ needs. This is especially important when potential consumers are customers, partners or others outside the organization.

Service consumption
To evaluate reusable services, service consumers should ask such questions as, “Are the requirements complete?” “Will a service fulfilling those requirements meet our needs?” “Are noted service-level agreements and quality-of-service figures appropriate?” Once again, tooling such as WebSphere Service Registry and Repository can assist as your services portfolio expands.

Testing SOA-based solutions
It is important to ensure that SOA project requirements are testable. As with any project, requirements drive test planning. SOA-based solutions require exceptionally rigorous testing because particular attention must be paid to external interfaces, security issues and service quality. Nonfunctional requirements should also be tested—particularly with small-scale services. Service consumers should ensure that expected functional behaviors, as well as error responses, are thoroughly tested. Tooling such as IBM Rational Service Tester for SOA Quality software can be used to execute the tests derived from the requirements.

It's essential that you make sure your SOA project requirements are testable, since many of your services have external visibility.
Definitive requirements for delivering successful SOA-based solutions.

### Highlights

**Do's and don'ts for SOA requirements**

**Do:**
1. Look beyond your traditional requirements techniques and consider more business-focused artifacts for your SOA projects.
2. Make sure you capture all the key nonfunctional requirements as early as possible, and keep validating these requirements as the project proceeds.
3. Move beyond the vision level to answer questions such as, “By how much?” and “By when?” when defining business goals as requirements.
4. Identify metrics and KPIs to help identify additional requirements on both large- and small-scale SOA projects.
5. Generate business use cases on large-scale SOA projects to provide a top-level view of the business from the outside—the outside is where your customers are!
6. Work with your test team early on in the lifecycle so team members can help create, validate and use the requirements.

**Don't:**
1. Assume you can go back to a proper requirements definition process on the next project.
2. Stop validating your requirements as business needs change—requirements definition is a continuous and ongoing activity.
3. Ignore nonfunctional requirements.
4. Limit your requirements to a service interface definition—you need more than that!
5. Try to use a system use case to document service functionality. A service invocation has to be stateless, but a system use case describes a strictly ordered, ongoing dialogue with a system—these are very different things.
6. Give up on requirements just because you’re moving to SOA-based solutions. Your business and your customers will hate you if you deliver the wrong solution.

While it’s not an exhaustive, this do’s and don’ts list can help you establish a good starting point for effective SOA requirements definition.
Definitive requirements for delivering successful SOA-based solutions.

Requirements for better business

Best practices for any software project emphasize the need to focus on requirements. But the need is even greater in SOA projects, where effective requirements definition and management are essential for achieving alignment between the solution and business goals.

To support successful requirements gathering and management, IBM offers process and automation tools designed to support best practices and provide linkages between requirements and other project artifacts. These tools can help you compose, gather, store, update, test and manage requirements throughout the service lifecycle. And just like it is on any complex software development project, a well-defined process is critical to efficiency, predictability and overall quality on SOA projects.
For more information
To learn more about IBM Rational software products for requirements management, contact your IBM representative or IBM Business Partner, or visit:

ibm.com/software/rational/offers/irm
The term SOA projects refers to projects creating SOA-based solutions.


Those developing solutions that require different service levels might also consider using IBM WebSphere Business Services Fabric software to define and manage policy-driven service levels.

This is a particular problem when the business process runs for a long time. How do we know whether it's safe to retire a service?

Good alignment to business goals often means a project will survive the budget cuts prevalent in IT departments if it can be shown to directly support business goals.

RUP includes content to assist with service-oriented modeling and architecture. This is based on a technique called IBM SOMA (Service-Oriented Modeling and Architecture) and focuses on linking services back to the business process, service identification, service specification and service implementation.

For details on these topics, review IBM RUP, Version 7.2 or later.


Service policies and their management are a significant topic in their own right. Service policies should provide agreed-upon criteria and constraints, should be centrally managed, and should be used to validate any services created or modified.