

SOA Governance

Achieving and Sustaining
Business and IT Agility

William A. Brown, Robert G. Laird, Clive Gee, Tilak Mitra

Foreword by Kerrie Holley



The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

© Copyright 2009 by International Business Machines Corporation. All rights reserved.

Note to U.S. Government Users: Documentation related to restricted right. Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

IBM Press Program Managers: Steven Stansel, Ellice Uffer

Cover design: IBM Corporation

Associate Publisher: Greg Wiegand

Marketing Manager: Kournaye Sturgeon

Publicist: Heather Fox

Acquisitions Editor: Katherine Bull

Development Editor: Ginny Bess

Managing Editor: Kristy Hart

Designer: Alan Clements

Project Editor: Chelsey Marti

Copy Editor: Keith Cline

Indexer: Erika Millen

Compositor: Gloria Schurick

Proofreader: Kathy Ruiz

Manufacturing Buyer: Dan Uhrig

Published by Pearson plc

Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U. S. Corporate and Government Sales

1-800-382-3419

corpsales@pearsontechgroup.com.

For sales outside the U. S., please contact:

International Sales

international@pearsoned.com.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: DB2, Lotus, Tivoli, WebSphere, Rational, IBM, the IBM logo, and IBM Press. Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. Microsoft, Windows, Windows NT, and the Windows logo are trademarks of the Microsoft Corporation in the United States, other countries, or both. Linux is a registered trademark of Linus Torvalds. Intel, Intel Inside (logo), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both. OSF/1 and UNIX are registered trademarks and The Open Group is a trademark of the The Open Group in the United States and other countries. ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office. IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce. Other company, product, or service names mentioned herein may be trademarks or service marks of their respective owners.

Library of Congress Cataloging-in-Publication Data

SOA governance : achieving and sustaining business and IT agility / William Brown ... [et al.].

p. cm.

Includes bibliographical references.

ISBN 978-0-13-714746-5

1. Web services. 2. Business enterprises—Computer networks—Management. 3. Computer network architectures. I. Brown, William, 1959 Oct. 20- II. Title: Service oriented architecture governance.

TK5105.88813.S63 2008

658'.05—dc22

2008042212

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671 3447

ISBN-13: 978-0-13-713084-8

ISBN-10: 0-13-713084-8

Text printed in the United States on recycled paper at Courier Stoughton in Stoughton, Massachusetts.
First printing December 2008

Foreword

Market forces drive the need for change and innovation in many companies. Whether the market force is mergers and acquisitions, changing market composition, regulatory demands, or new technologies, companies are finding that being competitive in demanding markets requires change; and, for many, IT can be an inhibitor to change. New ways of reaching customers and increasing flexibility is paramount where the expected benefit of flexibility is reduced time for system integration, increased asset reuse, and reduced time to market for solutions. Companies are responding in several ways, but overwhelmingly, the strategies that we see organizations adopting across the world means adoption of service-oriented architecture (SOA) in one or more facets.

The expected benefits of applying SOA vary from improved operational efficiency by deploying one or more shared functions across the organization to faster integration of core systems. SOA can reduce cycle times and cost for external business partners, facilitate flexible dealings with business partners, increase revenues, create new routes to market, and create new value from existing systems. SOA can reduce risk and exposure, improve visibility into business operations, or simply provide a more flexible business design by moving to architectures capable of business flexibility and game-changing business models.

These benefits cannot be achieved solely by defining and adopting a new architecture, strategy, or approach—that is, SOA. SOA software and services have quickly become a \$1.1 billion market. While line-of-business executives and CTOs are spending the dollars on an SOA, it is

up to line-of-business executives, business architects, IT architects, and designers to actually make sense of, construct, and implement an SOA that achieves the desired business goals and benefits. SOA governance is a critical enabler for achieving and sustaining desired business goals and benefits when adopting SOA.

One of the most difficult aspects of defining and implementing a new architecture is the definition and assignment of roles within an organization. SOA is no different; it continues to be viewed and defined depending on perspective and role.

Developers often equate SOA and web services as one and the same, whereas designers view SOA as a set of implementation patterns that should be applied during development. Many IT architects view SOA as another in a long line of architectural style. Executives (business and IT), line-of-business executives, and business stakeholders see SOA as a core underpinning to their IT strategy or as a critical enabler for business process management or key transformation initiatives. The point is that SOA means many things to different practitioners, highlighting the need for a shared vision and definition for SOA to achieve the promised benefits of SOA. So, how does an organization, a company, or line of business achieve the benefits of SOA when there is no common agreement within the organization?

For many organizations, SOA governance is a starting point for addressing this issue. Your first questions might be, “What is SOA governance, how does it help corporations reach this common agreement, and how can it make a difference? Why is SOA governance a critical success factor for both achieving and improvement in time to market for new products and features, reducing costs, improving business process flexibility, and ultimately turning business agility from a platitude to reality?”

The answer is not a short or easy one, but getting to that answer is easier if you read this book. The authors thoroughly address these key questions based on their collective experience of successfully implementing projects at organizations where business agility comes to life with the use of SOA governance. The authors recognize that not every organization is exactly the same, and so they provide the tools and guidance for how to make SOA governance work for any organization’s culture.

When I talk to practitioners and executives about SOA governance I often hear in their voices, see in their body language, and see in their eyes that familiar reluctance. I can almost hear them thinking, “Here we go again. Why do I need this esoteric thing called SOA governance? Our project does not have time to factor in issues around governance, and more important, it will not result in getting our projects completed faster.” I understand where they are coming from and nod affirmatively. Then, I ask a few questions about their reasons for adopting SOA, and I ask what their expectations are for applying SOA principles to their IT architecture. Most of the time, their answers center on reuse and the expected benefits of improved time to market, improved flexibility, and other competitive benefits. Then I ask them how the project will enable these goals and benefits. At this point, the room full of practitioners and executives tends to become very quiet. I go on to ask how flexibility is defined and how will they know when they have met their intended goals? As the discussion continues, the benefits of utilizing SOA governance, whether it be cost reduction, process flexibility, or improved fusion between business and IT, starts to become clear to everyone in the room.

I often tell a story of *déjà vu* for those who don’t know we have been down this road before. During the 1970s, when database technology was being introduced, we had these exact same conversations. Why do I need a data model? Exactly, what is a DBA? Let me understand this, I need a data modeler to do what? You are asking me to create data owners and have data stewards? Now we ask the questions about “services.” What is a service model? Why do I need service modeling? What is a service lifecycle? Who are service owners? This book addresses these questions.

If you are a practitioner, manager, or executive tasked with the job of making SOA benefits real in your organization, this book will help you take SOA governance from concept to reality. SOA governance describes both a framework and method for applying SOA governance in your organization and making it work given your organization’s idiosyncrasies. More important, the authors provide case study insights for an organization faced with the strain between strategic intent and tactical goals. The authors provide both tools and guidance for making SOA governance both practical and real.

This book is a must read for those who must understand and address all the challenges faced in making a SOA implementation successful. Authors Bill, Clive, Robert, and Tilak are seasoned practitioners with a wealth of experience in SOA deployments, and anyone who is tasked with making the benefits of SOA real will realize a competitive advantage from reading this book. I highly recommend this book and believe it makes an invaluable contribution to the IT industry.

Kerrie Holley
IBM Fellow
San Francisco, California
September 2008

Introduction

A Services Approach

“A man who does not think for himself does not think at all.”

—Oscar Wilde

Service-oriented architecture (SOA) has been around long enough to justify a close examination of the benefits from its adoption, adherence, and usage in any organization. It is about time that the promises of SOA are measured against real-world SOA implementation experiences and that a business be able to learn how to properly govern an SOA program without having to do it the hard way, like through the pain and expense of personal experience. Some of that is inevitable, but this book shows you ahead of time what to do to save your company time and money.

This chapter begins with a discussion of some key benefits of SOA. Based on our real-world experiences with numerous SOA initiatives and implementations, we then explain why so few organizations actualize the full benefits of SOA, why certain dimensions of SOA are not executed correctly, and how other aspects of SOA are implemented to maximize its advantages.

Benefits of SOA

“In all affairs, it’s a healthy thing now and then to bang a question mark on the things you have long taken for granted.”

—Bertrand Russell

SOA presents a compelling value proposition that addresses a distinct set of business challenges which enterprises face today. The fundamental tenet of SOA is that it demands as much of a priority and commitment from the business stakeholders as it expects from the IT department. SOA demands that for a business to be agile and adaptive, the organization must represent the enterprise’s core business processes through flexible business models. Next, one must expose its IT infrastructure, application capabilities along with its business critical information through a set of shared and reusable services so that each such service can participate in the implementation of the flexible business models. By building flexibility into the business models, through their representation as a set of participating services, enterprises can integrate third-party services into their core business processes, and thus reduce the cycle time and cost of integration with external businesses. SOA promises a way for businesses to model and structure their organizations and operations so that they can efficiently leverage individual capabilities, coordinate their efforts to leverage IT profitably, and reduce complexity. The net result is a greater ability for businesses to monitor, react, and adapt to new opportunities and changes.

SOA enforces standardized definitions and consistent implementations of business services, enabling IT transformations aligned with the business imperatives and drivers of the enterprise. The standards-based SOA layers between the digitized business processes and the underlying IT applications and infrastructure enable the business to confidently embrace agility, adaptability, and change through its flexible business processes. It also allows technology maturity, replacement, and modernization to take place without affecting or impeding the agility and adaptability of the enterprise business processes. Deconstruction and modularization of the business into a set of business-aligned services—a fundamental tenet of SOA—make SOA a close reflection of the business operating model.

The benefits of SOA are realized through the ways it tries to bridge the chasm that exists between the current business models and the traditional ones. As an example, a typical traditional business model is linear in structure. Business processes in such a traditional model are typically handled completely by individual departments or lines of business (LoB) in a company without any significant interaction and dependence on other parts of the organization. Today's business model must be nonlinear in nature, with parts of the business process typically being performed by either different departments in the organization or externally through business partners. For example, a customer may place an order through a company portal. Then, shared services in different parts of the organization, such as merchandising or the supply chain, perform different steps in the process. An example of this would be suppliers contributing to vendor-managed inventory or shipping being outsourced to a third party. This kind of desegregation of the business process, requiring a lot of flexibility, is made possible by SOA. The shared services are identified, designed, and implemented in ways such that a service can participate in multiple different business processes—both current and future.

SOA has many other benefits. By creating new value from existing systems, it changes our traditional view of legacy systems. It provides the value proposition of reusing legacy systems that have been the heart of existing enterprises and have been performing brilliantly. Reuse the existing system and applications without replacing them, but expose the capabilities that have been locked inside these proprietary and dated systems. As an example, consider an arcane Customer Relationship Management (CRM) package that stores business-critical customer-profile information. The access to such information typically requires a proprietary access protocol and data format. However, this customer information is increasingly required by businesses to provide, services such as, auto insurance, home insurance, and in the future, health insurance. Such critical data and functionality should be identified and exposed in a way that it can be accessed using a common and standard mechanism. SOA principles, best practices, and guidelines may be followed to achieve this goal. Such standard mechanisms of exposing business-critical information as a set of services enable an enterprise to drive down cost and operating expenses, eliminate duplicate systems through consolidation, and improve the time to market for new or enhanced capabilities. Through the incremental maturity of the enterprise service

portfolio, the cost benefits of SOA result in a financial boon to the enterprise. None of this happens by accident though; there needs to be an explicit governance capability—without which, such efforts are doomed to failure.

What Goes Right?

“Honest differences are often a healthy sign of progress.”

—Mahatma Gandhi

SOA is at the forefront of many business and IT initiatives. Consulting firms and product vendors have started to support SOA by demonstrating their capabilities, expertise, and maturity in this emerging discipline.

Organizations that have successfully crossed the initial hurdle of SOA have quickly realized that the most practical way to achieve the benefits of SOA's, reusability and flexibility is through standardization. In industries where a common domain model exists, such as ACORD in the insurance industry and eTOM in the telecommunications industry, enterprises have started to use those industry domain models as references. The industry domain model is usually inherited and customized to define an enterprise-level Common Information Model (CIM). The CIM is subsequently leveraged to derive the Common Message Model (CMM), which plays a critical part of the service definitions. The CIM and the CMM form the basis of standardization that can be applied to interfaces and data structures. Such standardization is also reducing the complexity of data-transformation requirements inside the enterprise perimeter—a necessary step to reduce the complexity of service-based integration.

There is now an increasing focus by companies to clearly understand the scope of the SOA transformation and follow well-proven methodologies for SOA.

SOA experts and consultants are keen to enforce a clear understanding of scope by employing a technique called goal-service modeling. The crux of this technique is in using business goals as the basis of defining and constraining the scope. Business goals are clearly articulated, prioritized, and signed off by the business stakeholders. So as to provide the greatest impact, only those services and applications are considered in

the scope of the transformation that have a direct traceability back to the realization of the prioritized set of business goals that the business is keen on achieving. Chapter 4 of the book *Executing SOA—A Practical Guide for the Service-Oriented Architect* has a detailed treatment of these techniques. (For more information about the book, see Appendix B, “References.”)

To mitigate and contain the risks related to learning this new, business-focused technique, organizations are performing a technical feasibility exploration (TFE). A TFE is performed in cases where the technology is not well proven and has a degree of complexity that is not easy to factor into a project plan based on a pure conjecture. TFEs are usually manifested through a set of proof of concepts (PoCs). These TFEs are identified and scoped ideally during the initial phases of the project, executed, and then their output and results are used to not only gain a better understanding of the technical complexity and nuances of service implementation but also to refine the architecture and design decisions. This added understanding assists in assigning resources (time, personnel, and expertise) commensurate with the complexity of the service implementations.

The industry has faced a nagging problem of how to crisply define the handoff between business and IT. Organizations are now keener to follow a prescriptive technique to document business requirements. Technology has assisted in formalizing this technique. We now have the luxury of using sophisticated business process modeling (BPM) tools that provide tooling and prescriptive guidance on how to capture the “as-is” and “to-be” BPMs. Many organizations have adopted the approach of using BPM tools and techniques to not only trace the lifecycle of the key business entities but also to document the requirements. We believe this is the correct approach to gain consensus and consistency of understanding and also to hand off requirements, in digital form, from the business to the IT departments.

Often, organizations in their early SOA adoption stages face daunting challenges in implementing new methodologies around SOA. These companies frequently face critical architecture decisions that risk large penalties for the wrong choices. These scenarios are addressed by working on a small piece of functionality that has an end-to-end coverage through all the layers of the SOA solution stack. Organizations use the outcome of such implementations to recalibrate the scope and complexity of SOA initiatives.

What Goes Wrong?

“He who has not first laid his foundations may be able with great ability to lay them afterwards, but they will be laid with trouble to the architect and danger to the building.”

—Niccolo Machiavelli

In the formative years of SOA adoption, which by no means are over, we have frequently seen organizations take their existing business and IT staff and assign them new roles that are required to execute on an SOA initiative. Application architects are assumed to play the role of the SOA architect, and traditional business analysts are assumed to seamlessly pick up the similar role in an SOA project. Such assignments expose a lack of understanding by organizations about the specific requirements of a role in the context of SOA. The decision makers assigned the responsibility to staff SOA projects need to understand and acknowledge that role transition is neither trivial nor simple. For example, you can't just transition an application architect to an SOA architect. The person who does this job needs a deeper knowledge of the new products and technologies that constitute the entire SOA solution. A paradigm shift must occur in the process of architectural thinking, with reusability perceived at a higher level of abstraction, that of a business service. The architect must identify services that have business relevance and traceability; therefore, the architect must soundly understand the organization's business goals and drivers. This transformation from one type of architect to an SOA architect is just one example of the new skills, expertise, and thought processes that need to be mastered to meet one role of an SOA adoption.

Traditional business analysts must transition to operate in an SOA environment. This paradigm shift requires them to use new tools, technologies, and methods to define and develop the business requirements. This helps to transcend traditional organizational boundaries and thinking in terms of reusability of services and business processes at the enterprise level.

Organizations often jump onto the SOA bandwagon without assessing whether the organization has the maturity to adopt SOA as a business and IT imperative. We recommend that organizations first assess their organizational, cultural, technological, infrastructural, and personnel capabilities. Organizations must understand where they are from an SOA maturity standpoint. Then, they can use the maturity indicator metrics to develop a plan that helps them iteratively progress up the SOA maturity ladder.

Organizations also tend to jump onto the SOA bandwagon without assessing their capability to adopt SOA as an enterprise initiative. Commonly, such organizations fail to reap the true benefits of SOA and thus doom their investment. To mitigate this risk, IBM, as one of the leaders in championing SOA, has developed the Capability Assessment Tool (CAT), which enables its customers to perform a self-assessment of their maturity as it pertains to SOA adoption. It is publicly available, and the underpinning of this tool is an IBM technique called the Service Integration Maturity Model (SIMM). SIMM is used to measure the maturity of a company's SOA across seven dimensions. It is used in SOA assessments to identify where a company resides in terms of maturity, what target state of maturity it wants to achieve, and the benefits gained from achieving that target state. (For more information about SIMM, see Appendix B.) We highly recommend organizations to use SIMM as a stepping-stone into the world of SOA to avoid the pitfalls discussed in this section.

Communication between the business stakeholders and the IT department is a key to the success of SOA. It is common to come across SOA projects that fail to develop and execute an effective communication framework that keeps the business stakeholders in the loop from start to finish. The business stakeholders should be the ones to finalize whether a service that is identified by IT makes business sense and is relevant for them. Failure to keep business stakeholders properly informed is a sure way to miss out on the true value proposition of adopting SOA.

In most companies, a major deficit is an efficient governance structure that incorporates and enforces an effective alignment mechanism between the business and IT. The continual fine-tuning of the services, based on business inputs, is a highly recommended approach to demonstrate how business agility can be achieved through IT agility underpinnings. Through continual and effective governance, tangible results and benefits of SOA can be achieved and demonstrated.

SOA projects are often poorly estimated. Currently, no de facto standard technique exists to estimate SOA projects. Therefore, it is advisable to be more conservative than aggressive regarding the estimated timelines for each phase of the SOA project. However, the reality is quite the opposite, especially in organizations where the SOA maturity is still in its infancy. Estimations are frequently fooled by the mantra that SOA reduces the overall latency between requirements and deployed code. Many projects, by falling into this trap, shortchange themselves after overpromising to deliver. The reality is that the advantages of SOA directly correlate with the maturity of the organization in their SOA adoption. For low- to medium-mature firms, project costs for SOA initiatives, for the first couple of projects, will typically be more than that of traditional component-based development. This extra cost—in the form of time, resources, and funding—should be factored into the estimation model.

The scope of SOA adoption must be broadened and considered from the viewpoint of the enterprise of which the business units are part. Executing individual SOA projects that have their own imperatives and are not aligned with or traceable to the business vision and drivers is a recipe for failure. To mitigate such risks, we recommend that an SOA Center of Excellence (CoE) be set up at the enterprise level. The members of the SOA CoE should be adequately represented by both the business and IT wings of the enterprise. One of the primary SOA CoE responsibilities should be to set up an SOA governance model and framework. The SOA governance model works as a central decision rights and management framework and ensures that any SOA project has an enterprisewide and cross-project scope. Coordinating IT project initiatives through a central governing body ensures the correct level of granularity of a service's focus: Each SOA project is a cog in the wheel of an enterprisewide SOA initiative; all cogs need to turn smoothly and in harmony for the wheel of enterprise SOA adoption to gain momentum and move forward.

Conclusion

SOA is an architecture paradigm that enables IT to organize, define, and implement their projects in such a way that their value can be directly correlated with the business goals, vision, and drivers of the enterprise. Success in this endeavor requires SOA Governance. Although the benefits of SOA have far-reaching and positive influence on how an organization can leverage its IT capabilities and assets to prepare for business agility and responsiveness, SOA adoption is a serious undertaking that requires the organization to define a structured, iterative, incremental, and phased approach. The approach should be commensurate with the maturity of the organization across its various disciplines and dimensions.

This book shares our collective experience and knowledge gained from numerous SOA projects in the practice of SOA Governance. As we discussed earlier, successful SOA projects share common “good” practices, whereas other SOA attempts fail to maximize the benefits possible or fail outright. Based on our experience, we believe that SOA attempts fail more often than they succeed at this point. With that in mind, our goal in this book is to teach you how to use SOA Governance to properly govern and manage the SOA journey, increase the return on investment of the results, and decrease the risks.

4

Governing the Service Factory

“Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the Universe trying to produce bigger and better idiots. So far, the Universe is winning”

—Rick Cook

Chapter 3, “Building the Service Factory,” discussed the value of establishing a service production line to automate SOA development activities, and outlined the tasks, roles, and responsibilities needed to establish such a service factory. Chapter 3 also described a set of work products whose adoption can implement practical SOA governance for the Plan & Organize and Program Management Controls domains introduced in Chapter 2, “SOA Governance Assessment and Planning.”

This chapter covers a practical approach for governing both the operation of the service factory and the management of services after they have been deployed to production.

Essential Competencies for Succeeding with SOA

An organization needs certain core technical competencies if it is to succeed with SOA. Good SOA governance is all about developing these competencies and creating governance mechanisms that ensure the service factory runs as smoothly and as efficiently as possible. The following subsections define these core competencies.

Effective Requirements Collection

The old adage “garbage in, garbage out” perfectly expresses the importance of capturing requirements correctly. It is simply impossible to build good software without precise functional and nonfunctional (i.e., technical) requirements. SOA has the major advantage that services should represent some clear agreement or contract between the consumers and developers. As long as good naming standards are used, the purpose of each service invocation (operation) should be instantly clear to both business and IT staff.

There is an important degree of judgment to be made in terms of the effort made in capturing and reconciling requirements from all prospective consumers of a new candidate service. In an ideal world, all potential consumers of each service will be canvassed for their specific requirements, and a service designed that accommodates the full set of needs expressed:

- This has the theoretical advantage of reducing any need for future reversioning of the service, and encouraging maximum reuse (both major positive goals).
- Unfortunately, it has the practical disadvantage of consuming a significant amount of analyst resources, and potentially adding complexity and cost to the development of the first version of the service.

The current balance of opinion is that some kind of 80/20 rule applies: 80% of potential consumer requirements can probably be decided by a skilled analyst with 20% of the effort of an exhaustive study. Defining optimum service granularity, service design, and quality of documentation is as important an enabler of future reuse as an exhaustive initial consumer and requirements analysis.

Creating and publishing a catalog of requirements is critical to encouraging reuse and avoiding unnecessary duplication. Creating a high-level business process model and a common glossary of business terms is a critical precondition to creating a requirements catalog.

An excellent practice is to define the nonfunctional requirements, together with criteria for testing and acceptance of the service, at the same time as the requirements are being captured.

Competency in Service Design

Although service architecture and technical details beyond the scope of this book, we need to make some comments in terms of governing the service design process:

- It is essential to have a formal set of critical architectural work products to guide service designers. These include architectural principles that must be followed, architectural decisions that need to be complied with, recommended design patterns, and a formal target production. A best practice followed by many large organizations is to create an architecture review board (ARB) that is responsible for maintaining the vitality of such architectural assets.
- A strongly recommended best practice is to encourage (ideally mandate) regular service design reviews that include as wide a range of technical talent as possible. This has the joint advantages of optimizing the design of each service, promoting consistency, ensuring compliance with standards and best practices, and, not least, helping to grow the architectural skills of the more junior team members through on-the-job mentoring.

Competency in Service Development

Modern software development kits (SDKs) or tools make the physical SOA development processes relatively straightforward, and using them promotes consistency of technical approach and high quality. In this chapter, we discuss some work products that can help to monitor and assist service development.

Competency in Service Testing and Deployment

A reputation for quality is hard to gain and easy to lose. Thorough testing and an efficient and error-free deployment process are essential to gaining and maintaining a reputation for quality. Again, we describe some work products to help govern these processes.

Competency in Operational Management and Monitoring of Services

One of the easiest ways to lose a reputation for quality is to have products that perform badly or that keep breaking down, so we also address this important area from a governance perspective.

Service Development Lifecycle Control Points

Most organizations already have some type of system development lifecycle (SDLC) and a methodology that is used to perform development, although we often see in practice a lack of enforcement of that approach across different business units, and even if a set of best practices, standards, policies, and patterns has been defined, they are not always enforced.

Effectively enforcing best practices and a consistent SDLC provides a reasonable entry point for real governance, while not being a huge stretch from what is already being performed via the SDLC. At the same time, if the governance maturity level of the organization can be increased to the degree that it is able to govern the SDLC, the organization is then in a much better position to proceed to the next phase of the SOA governance cycle and create program and organization governance.

The danger here for even initial attempts at SOA governance is that often some key individuals view the imposition of any process or governance as being something that might apply to other people but not to them personally. For them, it's an overengineered, useless exercise that just gets in the way of meeting their own deadlines. So, many governance processes are simply bypassed, or they're followed in a less than an enthusiastic manner. The main reason for this is that governance is imposed from the outside and the execution is onerous.

What would happen if governance were mostly automated, easy, and added value to the development process and actually helped with project deadlines? Would the skeptics be more willing to take the medicine if it genuinely eased their pain?

To adequately govern the SDLC, there is a need to establish measurements, policy, standards, and control mechanisms to enable people to carry out their governance roles and responsibilities as efficiently as possible, without introducing overly bureaucratic procedures.

Governance of the SDLC may be characterized by the sorts of decisions that need to be made at certain “control points” within the process of services development. A control point is a decision checkpoint that provides an opportunity to measure adherence to the established processes, whether you are on track to meet the targets and goals you have established, and then decide whether the way the processes are executed or managed needs adjusting. Knowing what decisions involved in the process are critical, when to make them, and understanding what measurements are needed to monitor those processes are all essential aspects of governance.

Certain activities within a process may be associated with a control point. At the end of each identified activity, there is a control point at which the governance function decides whether the program is ready to move to the next activity. Each of these milestones is a control point.

At its essence, the governance of the SDLC provides a way to identify control points and to define the governance rules. At each control point, it is necessary to identify the following:

- The roles for who does what at the control point
- The policies to be applied at the control point
- Measurements at each control point that should be applied and collected for later governance vitality actions
- The proof of compliance records to be created and archived

A control point will be created where there is a demonstrated advantage weighing the standardization and efficiency provided versus the time, effort, and possible project delay. The control point enables SOA governance the opportunity to ascertain progress, to communicate this progress, to forecast efforts for subsequent phases of the SDLC based on scope and issues found, to review and report compliance, and to facilitate the injection of expertise and qualified review of the artifacts, process, or decisions made by the development team.

Control points don't have to consist of huge formal meetings. Services and most automated business processes are smaller entities than projects, and there are many more of them. Therefore, the existing governance

approach has to be streamlined or it might grind to a halt. We've found in practice that effective control point reviews can be made during regular—typically weekly—sessions of a subset of the SOA enablement. A real productivity aid in performing these control point reviews is the use of previously completed *checklists*, signed off by one or more senior professionals as certification that one or more tasks has been completed successfully, and that the service, process, or other work product is fit for purpose and ready in all respects for the next task in the development process. These checklists should be viewed as contracts between different experts in the service development process. The most important part of the checklist is the signature block to show who exercised approval authority; people tend to be careful about the quality of anything that carries their personal reputation with it.

Another productivity aid is the use of automated tooling. As much of the governance control point as possible should be automated. This aids in better near real-time feedback to the developers and provides an easy method to recheck work that has been updated. In addition, human beings are busy and will tend to apply governance in an inconsistent manner. Machines are consistent but not usually as flexible as needed. The combination of the two provides an optimal governance mix.

Let's look at the control points needed to govern a generic development lifecycle, at least at a high level. Figure 4.1 represents a “governance dashboard” monitoring a typical SDLC with an eye toward the key concepts and the points where they must be addressed by SOA governance.

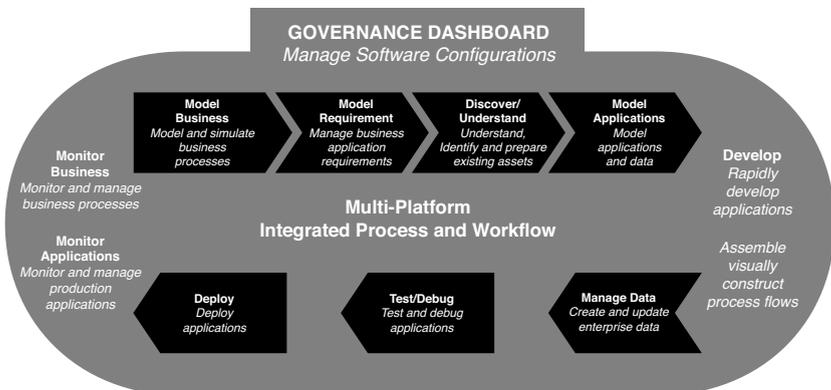


Figure 4.1 Software Development Lifecycle Governance Dashboard

As mentioned previously, we need a streamlined process that can handle the large number of services and automated processes that we need

to implement to have real impact on business agility and flexibility. However, that streamlined process must not sacrifice the *quality* of governance just because of the need for extra speed. That would be an unacceptable trade-off.

Some organizations deal with highly regulated processes that have mission-critical or life-critical products and need to apply highly formal, auditable governance to manage the risks involved. Other organizations have processes with lower associated risks that can be more lightly regulated. We have found in practice that the same governance process can handle both these extremes perfectly well. If there is a need for stricter governance, it can be met with tighter policies at the control points together with more stringent policy enforcement and compliance measurement. If less-strict governance is more appropriate, the same process can be used with less restrictive policies, fewer audits, and lower levels of checklist signoff required.

Even within a single organization, different processes may require different styles of governance. Some processes, such as service certification, require stricter governance than other processes, such as solution architecture. Different organizational cultures require different levels of autonomy in decision making. Good governance requires good judgment.

First, let's update our Figure 4.1 with the location of these control points so that you have a visual representation in mind as you read their descriptions. Figure 4.2 shows where the control points occur in that development cycle.

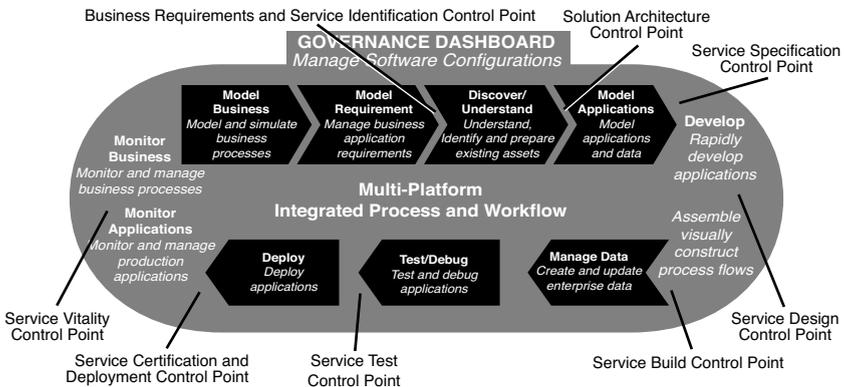


Figure 4.2 Software Development Lifecycle with SOA Control Points

Here are descriptions of these control points.

Business Requirements and Service Identification Control Point

For an SOA approach, there is an emphasis on creating services that provide agility and reuse for the business. This first business requirements and service identification control point consists of a high-level review to determine that services are being identified in accordance with the services selection and prioritization policies that were described in Chapter 3, “Building the Service Factory.”

This first business requirements and service identification control point should address the following types of questions:

- **Business goals.** What are the business goals that the business seeks to attain and how do we measure the benefits or progress toward the business goals via key performance indicators (KPIs)?
- **Do the requirements as we currently understand them clearly support those goals, and do they align with the business heat map described in Chapter 3?**
- **Are those requirements sufficiently understood and agreed to? Are they presented in a form such as use cases, business process models, sequence diagrams, or class diagrams that are consistent with the SOA development approach?**
- **How do we provide traceability of the requirements so that we can ascertain that those requirements have been met during the development process? Have those requirements been entered into an enterprisewide requirements and business rules catalog? Is there any conflict with existing entries in that catalog?**
- **Which of those requirements could be translated into good candidate services, either because they represent functionality that may be needed by multiple consumers or that might be needed for process automation? Which requirements could be better supported by deploying applications, automated processes, or manual processes?**
- **Where we have identified candidate services, have we identified potential consumers, and determined whether any of them have specific requirements that should be considered?**
- **Given finite IT resources, what development priority should we assign? Is ownership of any new candidate IT asset defined, and is outline funding available for its development?**

Solution Architecture Control Point

Different IT developers and groups, if left to make all design decisions on their own, would invariably use completely different platforms, coding languages, tools, styles, methods, and techniques. This variation adds cost and complexity to the ability of the business to make future changes, and makes future maintenance very hard and costly. Further, it reduces the reliability, stability, and interoperability of the organization's IT assets. We have seen this problem at many organizations that we have visited. Simply put, the purpose of the solution architecture control point is to prevent that expensive multiplicity of approaches from occurring ever again.

Essentially, any proposed IT artifact that makes it past this control point is part of the IT build plan. For the area of solution architecture, the governance should control for a series of criteria the following:

- Do the proposed standards, policies, and reference architectures—the solution architecture—identify the standards, policies, and design patterns to be followed in the service implementation? This will include reference architectures, platform standards for hardware, and software-usage standards.
- Have any reusable assets been identified and assessed for suitability? Has the service sourcing policy been followed?
- Have the nonfunctional requirements been identified and assessed? This includes the number of transactions per time unit, a busy hour analysis, the service performance required, presentation access to the service functionality, data managed by the service, space required for the installation of the service, and any dependency and configuration requirements. Governance must validate that all these are considered and addressed.
- Governance must validate that all security policies are being considered and addressed.
- Governance must validate that all legal and regulatory policies are considered and addressed.
- By this stage in the development of IT assets such as services or automated processes, the technical IT staff involved should have a pretty good idea about the complexity of the tasks involved, and the probable level of resources required to complete development. Should development of the asset be confirmed, the scope reduced, or the asset abandoned?

Service Specification Control Point

A service specification should be created for each service whose development has been approved. Best practices for service design must integrate both an IT and business perspective for the design of the interface and the responsibilities of each service. Because the service specification is, in effect, the organization's face to business partners, customers, and other stakeholders, the *service externals*—those details of a service that are to be made public—become an important part of the overall business design. The design should take into account the requirements of all potential service consumers (within reason), and be created at a granularity that maximizes business value. For the area of service identification and specification, the governance should control for a series of criteria the following:

- Does the service identified make sense, is at the right granularity, and is not duplicating an existing service?
- Does the service specification follow all SOA standards and policies?
- Does the service specification follow the messaging model? If not, should an exception be granted?

Service Design Control Point

After the service solution architecture has been turned over to the design team, a number of design elaboration decisions must be made. Collectively, these form the *service internals*—a set of design models, notes, and advice that will guide the service developers as they create and test the service code. For the area of service design, the governance should control for a series of criteria the following:

- Has a service architect confirmed that the design should be able to meet the nonfunctional and functional requirements for this service?
- Have the service designer and data architect agreed that the service can be made to conform to the signature (that is, inputs and outputs) described in the service externals?
- If a service is wrapping an existing or planned application, are the necessary interfaces to that application well defined and stable (that is, won't change if a new version of that application is installed)?

- Have the monitoring metrics (for example, usage, quality of service [QoS] levels) been established?
- In the case of automated processes, have the monitoring requirements been defined and planned?
- In the case of long-running automated processes, have all the necessary actions to handle recovery from process errors or technical failures been addressed?
- Is the overall quality and level of completeness of the service specification package good enough that the service developers or process developers can complete development without further input?

Service Build Control Point

After the service design has been turned over to the service build team, a number of implementation decisions need to be made before development of the code or executable model. In the interests of consistency and quality, we strongly recommend the use of code walkthrough reviews, where peers (that is, other service developers or process developers) review the work in progress and offer constructive criticism.

The service build control point is effectively the last of these code walkthroughs, and should be performed with slightly more formality than the others. Questions that should be addressed include the following:

- Was the asset coded in accordance with the design?
- Does the code follow the accepted coding standards?
- Have all the associated artifacts (for example, load libraries, metadata files, resources) been defined to create a transportable build? Have the versions of each of those artifacts been checked to see that there are no version conflicts with services already in production?

Service Test Control Point

Service testing is different from testing complete IT solutions or applications. Because services and automated processes do not have their own user interface, it is not possible to perform user acceptance testing directly on services or automated processes. Code frameworks or specialized tools are needed to exhaustively test services and automated

processes thoroughly to avoid uncovering problems during later formal user acceptance testing when the rest of the IT solution that uses those services or processes has been completed.

SOA governance must ascertain that the services test is being performed in a manner conducive to a services approach, and that exhaustive functional and nonfunctional tests have been passed before releasing any SOA asset to production. The service test team must create and use the right service test environment with tools and data to affect a comprehensive test. This should include the following:

- Using the optimum set of service test tools and frameworks.
- The use of an automated build and test environment that can enable fast changes of the tested software and regression testing. This environment must closely resemble the production environment.
- A load/stress test tool to test nonfunctional requirements, specification, creation, and loading of realistic but artificial test data.
- A test management reporting tool to keep management apprised of the testing status.
- Trace the test case to the original user requirements.

Service Certification and Deployment Control Point

The objectives of the deployment are to migrate the services to the production environment while minimizing client downtime and impact on the business. This process is subject to many errors if performed manually. It is vital that the correct version of the services be deployed and that any deployment binding with other services and applications be performed quickly and correctly. Areas for governance to validate include the following:

- The use of a tool that automates the deployment and back-out process.
- Final certification checks have been made against the services to verify compliance with all policies and standards and being able to demonstrate that what was tested matches not only the requirements but what was delivered, and that no corrective changes made during testing have invalidated other test results.

- IT operations have completed acceptance testing and have formally accepted the asset, signifying their confidence in being able to operate it within the terms of the QoS specified for it.
- The service registrar and business service champion have reviewed the service description in the service registry and approved it.

Certification of a service or automated process is a formal “passing out” ceremony, and granting of certification should signify that the SOA enablement team is happy for their reputation to be associated with the performance of the new asset.

Service Vitality Control Point

Service vitality takes place periodically as part of SOA governance to check up on and update the governance processes, procedures, policies, and standards in reaction to the results of the real world. This involves examining any and all lessons learned in any of the SOA planning, program control, development, or operations activities. It also includes such things as comments and feedback from all stakeholders and an examination of any common patterns (for example, common exemption requests or common reasons for failure to pass one or more control points) that need remedial action. Metrics in the efforts required for each stage of the development process can show trends that indicate improvements or declines in their vitality.

A formal service vitality control point review should be conducted every three to six months to determine whether the SOA transition remains on track, and whether the level and style of governance is optimal.

Individual service or automated processes should be reviewed every 6 to 12 months. Usage data of all versions of each service can determine any “stale” versions that can be deprecated or deleted, and whether the deployment options taken and decisions on who should own and who should access each service are optimal.

The Service Development Domain

“Another flaw in the human character is that everybody wants to build and nobody wants to do maintenance.”

—Kurt Vonnegut

The Service Development domain covers the lifetime of services or automated processes from an original concept through to deployment into production. This section covers advice on governance of this development domain. Chapter 6, “Managing the Service Lifecycle,” looks in a little more detail at governing specific steps with the service lifecycle.

Most IT departments we have visited currently spend the majority of their resources on system maintenance. One reason for this is that most IT systems consist of applications where there is generally no clear separation between business logic, graphical user interface (GUI) management, and workflow. Any change to an application of this type risks damaging all of these three aspects, and maintenance becomes labor intensive without good design documentation—which is more often than not a major issue.

SOA makes a clear distinction between business logic and workflow. SOA is all about packaging pure business logic as reusable transactional services, and using executable models to manage workflow. SOA does not become involved with any particular style or fashion of GUI or GUI control. In a fully SOA-aligned world, humans will make requests to IT systems to perform tasks, make decisions, or initiate business processes using whatever GUI is best suited to that job; automated business processes can in turn initiate tasks by invoking services or by requesting human intervention or involvement.

In this world, IT solutions grow incrementally as new business processes become automated, and interactions with third parties are increasingly performed using service executions without direct human intervention. Whenever new business initiatives, market forces, or governmental regulations require changes to operating procedures, new or revised automated business processes can be constructed rapidly by reworking or extending existing automated processes. We can expect

technology to continue to change rapidly, so the users can interact with the IT systems in radically different ways in the future. Because the SOA approach insulates the user interface from the physical implementation of business function, we can expect that most of the services we create now will probably still exist in some form for the next 20 years or more, whatever interface is used to access them.

End-to-end service development is a complex process. Planning and governance has to be meticulous because there are many complex tasks with many interdependencies. There are critical tasks for governing service development that the governance specialist needs to address in the form of a transition plan that is created and worked just like any other project plan.

Key Capabilities Needed to Govern Service Development

The Service Development domain is highly complex and involves technology that is changing rapidly. Governing such complexity requires a high level of organizational and political skills, together with good knowledge, not just of the SOA technology itself, but how to use it optimally. Few individuals can combine those skills at a world-class level, so practical governance relies on close cooperation between some key technical and nontechnical roles. Close collaboration, and complete trust, among the SOA governance lead, the business service champions, the lead SOA architect, and the lead service architect is critical to success.

Table 4.1 describes the capabilities that the organization needs to govern service development, and recommends some key work products that can assist this complex activity. It is organized in the same way as Table 3.1 in Chapter 3.

Table 4.1 Service Development Domain—Capabilities, Risks, and Remedial Work Ducts

Capability	Associated Issues and Risks	Risk Level	Governance Work Products	Cost of Remedy
1. Services development lifecycle controls	Need to have a streamlined reliable end-to-end development process that finds issues and problems early and corrects them immediately. IT resources should be used as efficiently as possible.	Critical	<ul style="list-style-type: none"> ■ SOA development approach ■ Service build plan ■ Service construction estimation metrics ■ Service construction monitoring plan ■ SOA development performance report ■ SOA reuse and ROI report ■ SOA development lessons learned ■ Control point minutes ■ SOA governance policy exemption ■ Service design walkthrough notes 	High

Capability	Associated Issues and Risks	Risk Level	Governance Work Products	Cost of Remedy
2. Requirements gathering and prioritization	<ul style="list-style-type: none"> ■ No IT solution will be of any value unless the requirements are accurate; need to keep requirements as simple as possible, and avoid unnecessary embellishments ■ There is a risk of embarrassment/punitive fines if governmental regulations are ignored. ■ Business process models should be incrementally developed into executable models. ■ There must be no security violations that threaten data integrity or confidentiality. 	Critical	<ul style="list-style-type: none"> ■ Functional and nonfunctional requirements ■ Enterprisewide requirements and business rules catalog ■ Functional and nonfunctional requirements checklist ■ Regulatory compliance approach ■ Regulatory compliance checklist ■ Executable modeling approach 	High
3. Service identification	<ul style="list-style-type: none"> ■ Different potential consumers may have incompatible requirements. ■ Resources will be wasted if incorrect choices are made between delivering services or applications. ■ Functionality should be sourced in the most cost-effective way; sometimes this will be using a commercial package that meets most of the requirements, sometimes it is necessary to develop solutions in-house. ■ Need to define services of correct granularity to maximize their reuse. 	Critical	<ul style="list-style-type: none"> ■ Service granularity, visibility, and accessibility checklist ■ Service reusability guidelines ■ Service sourcing policy 	Moderate

Table 4.1 Service Development Domain—Capabilities, Risks, and Remedial Work Ducts (continued)

Capability	Associated Issues and Risks	Risk Level	Governance Work Products	Cost of Remedy
4. Service specification	<ul style="list-style-type: none"> ■ Need to ensure QoS designs and documentation to avoid costly rework after flawed development. ■ Good service design can minimize the need for managing multiple service versions. 	Critical	<ul style="list-style-type: none"> ■ Service specification ■ Service specification checklist ■ Service design checklist ■ Service security approach ■ Service security checklist 	Moderate
5. Service realization	<ul style="list-style-type: none"> ■ Effective software development tools can markedly increase productivity and accuracy. ■ Services should be exhaustively tested to ensure they meet all functional and nonfunctional requirements. ■ Bugs are costly to correct and embarrassing; zero errors is not an impossible target if testing is thorough. ■ An efficient configuration management/build process is vital to efficiently transfer code between development, functional testing, performance testing, preproduction, and production environments. 	Critical	<ul style="list-style-type: none"> ■ SOA development tools ■ Service build management approach ■ Service security checklist ■ Service version management approach ■ Service test plan 	High

Capability	Associated Issues and Risks	Risk Level	Governance Work Products	Cost of Remedy
6. Service certification	<ul style="list-style-type: none"> ■ Testing services may require connectivity to systems and data that closely mimic operational systems, such as servers, mainframes, and realistic but artificial test data. ■ Any problems that occur after deployment will damage the reputation of the SOA enablement team. ■ Instituting a formal certification process helps to establish the enterprise's commitment to quality and value. 	Critical	<ul style="list-style-type: none"> ■ Service acceptance checklist ■ Service deployment approach ■ Service deployment checklist ■ Service level agreement ■ Service version management approach ■ Service test results 	Moderate

Service Development Domain Work Product Definitions

In the preceding section, we introduced a number of work products that have been proven to be effective in helping to govern successful SOA transformations. Although we will describe them as if they are individual text documents or diagrams, in practice, many of the technical work products are best managed as models within tools specifically designed for manipulating them. Models of service internals, for example, should be edited and distributed using a specialized tool that can visualize and validate the interactions between the service components, and then generate most of the service code. No text document or simple picture can do that.

Documenting requirements involves many subtleties and creates multiple opportunities for misunderstandings to occur. A process model contained in a specifically designed process modeling tool that ensures logical continuity, and that requires all possible logical branches to be handled, is much more likely to be interpreted correctly than a use case written as a textual document. Better still if the modeling tool supports full emulation of the process steps.

Many of these work products represent intermediate deliverables that are passed from one professional to another on the service production line. These artifacts must be seen as representing a formal contract between the work product producer and the work product consumer. Delivery of the work product should formally signify that it is of the necessary quality and degree of completeness, and delivery of a substandard work product should be seen as a serious breach of contract. The control points described earlier in this chapter represent checkpoints to assess the quality of the key work products.

Some of the work products that can help govern the Service Development domain have already been described in Chapter 3, where, in addition, the roles of those involved in their production were defined. When creating the SOA governance plan, the SOA governance lead will need to select and create the work products that best manage SOA development service development activities without detracting from the ability to get projects done on time. Some of these work products have already been defined in the preceding chapter. Here are the descriptions of those new work products you should consider adopting, in alphabetic order.

Control Point Minutes

Description: These serve as a record of the results of the key SOA governance reviews, and can provide useful input to any systemic issues and the vitality of the SOA governance approach. The SOA governance lead or program management office (PMO) should examine these minutes to look for any common patterns of issues that need to be addressed by the creation of additional architectural decisions, design standards, best practices, or additional training.

Purpose: Ensure continued vitality of the SOA development approach.

When needed: Should be completed immediately after any service control point.

Responsible: Control point reviewers, PMO.

Accountable: SOA governance lead or PMO.

Consulted: Control point reviewers.

Informed: SOA enablement team.

Executable Modeling Approach Work Product

Description: This defines standards and best practices for developing executable business models, helping to ensure their consistency and quality.

Purpose: Define how automated business processes will be created.

When needed: Create this as soon as the SOA development approach work product has been approved.

Responsible: Service architects, process modelers, process developers, business analysts, monitoring developer.

Accountable: Lead SOA architect or lead service architect.

Consulted: SOA enablement team.

Informed: Process modelers, process developers, monitoring developer.

Functional and Nonfunctional Requirements Work Products

Description: For each service, these define exactly what each operation does, and how well it is expected to do it. Nonfunctional requirements include such factors as performance, availability, systems management, and security. There will generally be a fairly standard set of nonfunctional requirements that apply to most services, and individual services should have to define only additions and exceptions to them. Targets such as latency should be defined in measurable terms; for example, “95% of individual executions of this operation should have a latency of less than 1 second, to the glass on a portal screen, and 90% of executions should complete in less than 1.5 seconds.”

As already stated, a good practice for cataloging functional requirements is to maintain a requirements and business rules catalog in a database, indexed by the corresponding business entity in the enterprise data model (EDM) or messaging model (MM). This can avoid wasted effort in duplicating requirements analysis. Requirements should be classified into Mandatory, Valuable, and Optional. In the case where requirements are specific to a single line of business (LoB), this should be made clear in the text describing the requirement.

Purpose: Good management of requirements can avoid duplication of effort and help enable reuse.

When needed: A standard set of nonfunctional requirements should be created as soon as the SOA development approach work product has been

approved; functional requirements should be assessed at the business requirements and service identification control point, and nonfunctional requirements should be assessed at both solution architecture and service design control points.

Responsible: Service architects and business analysts (for nonfunctional requirements), business analysts and process modelers (for functional requirements).

Accountable: Business service champion.

Consulted: All service architects, business analysts, process developers, service developers.

Informed: SOA enablement team, QA, testing team.

Functional and Nonfunctional Requirements Checklist Work Product

Description: This checklist is used at the business requirements and service identification point, and nonfunctional requirements should be assessed at both solution architecture and service design control points to ensure that all potential consumers are satisfied that their individual requirements have either been met or have been explicitly ruled out of scope in the current planned service release, and that the design approach for this service should be able to satisfy the NFRs and SLA terms and conditions.

Purpose: Ensure the highest possible quality of services.

When needed: Functional requirements should be assessed at the business requirements and service identification control point, and nonfunctional requirements should be assessed at both solution architecture and service design control points.

Responsible: SOA governance lead creates the template, lead business analysts and service designers complete the checklist for individual services.

Accountable: Business service champion.

Consulted: Business analysts.

Informed: SOA enablement team.

Regulatory Compliance Approach Work Product

Description: Specifies what regulations apply under which circumstances (for example, the country in which the service requestor is based), together with the processes that will be followed to ensure compliance.

Purpose: Avoid the embarrassment and potential punitive consequences of disobeying applicable regulations.

When needed: Should be created as SOA development approach work product has been approved.

Responsible: SOA lead architect, PMO, business service champion, security architect.

Accountable: SOA executive sponsor or existing IT governance function.

Consulted: SOA enablement team.

Informed: PMO.

Regulatory Compliance Checklist Work Product

Description: The template for this document is created by the security architect and any existing IT governance group, and then approved by the lead SOA architect. Individual instances of this checklist are completed by the service designer and approved by the security architect. The regulatory compliance checklist for a given service should be endorsed at multiple control points to ensure that no changes have been made to the service that might invalidate the integrity of its security.

Purpose: Avoid the embarrassment and potential punitive consequences of disobeying applicable regulations.

When needed: Should initially be prepared in time for the service specification control point, then re-reviewed at the service build, service test, and service certification and deployment control points.

Responsible: SOA lead architect, PMO, service designer, service architect.

Accountable: Security architect.

Consulted: SOA enablement team.

Informed: PMO.

Service Acceptance Checklist Work Product

Description: A checklist used to confirm that IT operations agree that it can successfully operate and manage a specific service within the terms of its SLA.

Purpose: Ensure the highest possible QoS.

When needed: Should be completed before the service certification and deployment control point, for which it is a prerequisite.

Responsible: IT operations.

Accountable: IT operations.

Consulted: Service architect, service designers, service testers.

Informed: SOA enablement team.

Service Build Management Approach Work Product

Description: The approach to build management must ensure that SOA components can be migrated easily among development, testing, preproduction, and production environments repeatedly, without error or omission.

Purpose: To maintain continuity of operation, it is essential that the approach to build management is error free, and that it is always possible to re-create any given build configuration.

When needed: Should be created as SOA development approach work product has been approved.

Responsible: SOA lead architect, existing build manager.

Accountable: SOA lead architect.

Consulted: IT operations.

Informed: Service developers, process developers.

Service Build Plan Work Product

Description: This is the set of candidate service operations and automated business processes that comprise the formal SOA asset construction plan. This work product is an essential tool for managing SOA development in that

- It defines the results of the service prioritization.
- It's a key control document for service development project management.
- It communicates the SOA development plan and status to the rest of the organization.

Note that services, in general, have multiple operations—for example, a Customer service may have such operations as lookupCustomer and modifyCustomer. Not all operations will have the same development priority; there is no need to create all operations of each service at the same time.

Purpose: Manage the service development priorities and communicate those priorities to the stakeholders.

When needed: Should be created as soon as any services are planned, and it should be updated weekly.

Responsible: Lead service architect, service registrar, SOA business analysts.

Accountable: Business service champion.

Consulted: PMO, service architect, service developer service designer, process modelers, process developers.

Informed: SOA enablement team, PMO.

Service Construction Estimation Metrics Work Product

Description: These are the resource metrics captured as a result of the service construction monitoring plan. They provide invaluable input to the project management of new services and SOA projects. A steady improvement in these metrics is a sign of growing SOA maturity and effective SOA governance.

Purpose: These are important metrics to enable effective governance of the service factory.

When needed: Should be created as soon as any services are planned and data is captured for every service that is created.

Responsible: Lead service architect, SOA governance lead, PMO.

Accountable: Lead service architect.

Consulted: SOA enablement team, PMO.

Informed: PMO.

Service Construction Monitoring Plan Work Product

Description: This is the plan to capture and continuously update metrics on the resource efforts needed to complete each step in the service and automated business process lifecycle. Typically, resource estimates for each construction step are grouped into complex, intermediate, and simple services/processes.

Purpose: These are essential metrics to enable effective governance of the service factory.

When needed: Should be created as soon as any services are planned and data is captured for every service that is created.

Responsible: Lead service architect, lead SOA architect, SOA governance lead, PMO.

Accountable: Lead service architect.

Consulted: SOA enablement team.

Informed: SOA enablement team, PMO.

Service Deployment Approach Work Product

Description: The current IT deployment approach should be adjusted to allow for the characteristics of services. Typically, the number of deployments cycles in a traditional IT development approach is limited to a small number of major increments each year, to maintain stability of the operational systems, whereas the deployment of services can be almost an everyday occurrence.

Purpose: Ensure that services are deployed in the optimum fashion at the most advantageous times.

When needed: Should be created as SOA development approach work product has been approved.

Responsible: IT operations.

Accountable: Lead service architect.

Consulted: IT operations.

Informed: Service developers.

Service Design Checklist Work Product

Description: This checklist is used to ensure that the service internals contained in the service specification work product are high standard, complete, and unambiguous.

Purpose: Ensures the highest possible quality of services and reduces the need for rework.

When needed: A checklist template should be completed as soon as the SOA development approach has been approved. Instances of this checklist should be completed for each service before the service design control point.

Responsible: Service architect, QA, security architect create the template; service developers complete individual checklists.

Accountable: Service architect.

Consulted: Service designer, service developer.

Informed: SOA enablement team.

Service Design Walkthrough Notes

Description: These are relatively informal notes that describe the results of service design walkthrough sessions (for example, “chalk and talk” working sessions where service architects, designers, and developers meet to discuss the design of individual services, both to optimize that design and to mentor the less experienced staff members).

Purpose: Ensure continued vitality of the SOA development approach by identifying any common concerns that might require the creation of additional architectural decisions, design standards, best practices, or additional training.

When needed: Should be completed after every service design walk-through session.

Responsible: Service designers, service developers.

Accountable: Service architect.

Consulted: Lead SOA architect.

Informed: SOA governance lead.

Service Deployment Checklist Work Product

Description: This is a checklist used to ensure that services are deployed in the fashion described in the service deployment approach, based on a template created as part of the service deployment approach. It should take into account the fact that some services may have multiple instances, multiple access channels, and different QoS levels for different categories of consumer, according to the requirements specified by the service architect or service designer.

Purpose: Ensures that services are deployed in the optimum fashion.

When needed: The service deployment checklist should be completed before the acceptance process.

Responsible: IT operations creates the checklist template with assistance from service architects. Service developers and service architects fill in a checklist for each individual service.

Accountable: Service architect.

Consulted: IT operations.

Informed: Service developers, service registrar.

Service Granularity, Visibility and Accessibility Checklist Work Product

Description: This checklist is used to ensure that services have the optimum scope and are visible and available to any potential consumer that may need to access them.

Purpose: Ensures the highest possible QoS, with maximum business value and reuse potential.

When needed: Complete this before the solution architecture control point and revalidate it at the service design control point.

Responsible: SOA governance lead creates the template, and lead business analysts and service designers complete an individual checklist for each service.

Accountable: Business service champion.

Consulted: Business analysts, service registrar, service designers.

Informed: SOA enablement team, PMO.

Service Level Agreement Work Product

Description: SLAs represent the terms and conditions of a contract among service consumers and service providers. IT operations is responsible for monitoring that both parties comply with all their terms and conditions. SLAs should include the following:

- Guaranteed QoS levels, based on the corresponding service nonfunctional requirements
- Technical support terms and conditions (hours of operation of the help desk, incident response times by problem urgency, and so on)
- Constraints on the service consumer (specifying, for example, that QoS cannot be guaranteed if the service consumer exceeds a given threshold in the rate of service requests)
- For any services that are being offered for a fee, either internally or externally, the pricing structure of service requests
- How version management will affect consumers such as how many versions of services will be supported, length of time support will be available for deprecated services, how much warning service consumers will have of changes, or service retirement

Purpose: Define contractual terms for service usage.

When needed: Standard SLA terms and conditions should be established by the SOA governance lead as soon as practicable.

Responsible: SOA governance lead, business service champion.

Accountable: Business service champion.

Consulted: SOA enablement team, PMO.

Informed: Service owners, service consumers, PMO.

Service Reusability Guidelines Work Product

Description: Based on the SOA development approach, this work product contains guidance to service designers on how to design services with maximum reusability.

Purpose: Ensures that services are designed so as to maximize their reuse potential.

When needed: Should be created as soon as the SOA development approach work product has been approved.

Responsible: SOA lead architect, service architect, SOA governance lead.

Accountable: Business service champion.

Consulted: Business analysts, process modelers.

Informed: SOA enablement team, QA.

Service Security Approach Work Product

Description: The optimum approach for implementing SOA security to enforce the following:

- Authentication and authorization of all service consumers
- Nonrepudiation of service execution requests
- Protection of enterprise data assets
- Encryption of all data transmitted over channels such as the Internet

Purpose: Create and maintain a secure SOA production environment.

When needed: Should be created as SOA development approach work product has been approved.

Responsible: Security architect, lead service architect.

Accountable: Security architect.

Consulted: Operations management.

Informed: SOA enablement team.

Service Security Checklist Work Product

Description: Individual instances of this checklist are completed by the service designer and approved by the security architect. The service security checklist for a given service should be endorsed at multiple control points to ensure that no changes have been made to the service that might invalidate the integrity of its security.

Purpose: Ensure that services that do not comply with the requirements of the service security approach are never deployed into production.

When needed: Should initially be prepared in time for the service specification control point, then re-reviewed at the service build, service test, and service certification and deployment control points.

Responsible: The template for this document is created by the security architect and approved by the lead service architect; individual checklists are completed by a service developer and service tester, and then approved by a security architect.

Accountable: Security architect.

Consulted: IT operations.

Informed: SOA enablement team.

Service Sourcing Policy Work Product

Description: Based on the SOA development approach, the service sourcing policy describes how, once the need for a specific service or service operation has been established, the organization should choose between the options:

- Develop the service functionality from scratch.
- Use an existing IT asset or application, wrapped as a service.
- Buy a commercial, off-the-shelf (COTS) product that is or can be wrapped to be exposed as a service.
- Subscribe to an external software as a service vendor who performs that activity.

Purpose: Ensure consistency of approach in buy versus build decisions.

When needed: As soon as the SOA development approach has been approved.

Responsible: SOA governance lead.

Accountable: Procurement.

Consulted: Business service champion.

Informed: SOA enablement team, PMO.

Service Specification Work Product

Description: The service specification is not a single document, but rather a package that contains all necessary information needed to use, build, test, and certify that service or automated process. It should include the following:

- The service externals and service internals discussed earlier.
- Details of any interfaces to existing IT assets that the service will need to access.

- Functional and nonfunctional requirements for the service.
- Security information for the service (who can access it, authentication, encryption, nonrepudiation).
- Deployment options (for example, need for geographic diversity; “platinum,” “gold,” “silver” levels of service support)
- Monitoring and systems management requirements.
- Test plan and test data.
- The acid test of the level of completeness and accuracy of a service specification package is that it contains no ambiguity and that it provides a reasonably skilled offshore developer with all the information necessary to develop a high-quality code deliverable without the need for asking any questions to clarify any item it contains. For this purpose, models contained in sophisticated tools are much more valuable than words and pictures in most cases.

Purpose: Ensure the highest possible QoS.

When needed: Service externals need to be completed before the solution specification control point and the whole service specification completed before the service design control point.

Responsible: Lead service architect, QA, SOA governance lead create the templates; the service designer and service architect complete each service specification package.

Accountable: Business service champion.

Consulted: SOA enablement team.

Informed: Service consumers have access to the service externals, and only the SOA enablement team (specifically service developers and service testers) should be given access to service internals.

Service Specification Checklist Work Product

Description: This checklist is used to ensure that all service specification packages are created to the highest possible standards, contain all the information needed, and are clear and unambiguous.

Purpose: Ensure the highest possible QoS.

When needed: Should be completed before the solution specification control point.

Responsible: Service designer, QA, and service registrar.

Accountable: Business service champion.

Consulted: SOA enablement team.

Informed: SOA governance lead.

Service Test Plan Work Product

Description: Thorough functional and nonfunctional testing of services is vital to ensuring their quality. For each service, this plan should describe all the tests that need to be carried out, and what the expected result should be.

Purpose: Ensure the highest possible QoS.

When needed: Should be completed immediately after the service design control point.

Responsible: Service test manager, business analysts.

Accountable: Business service champion.

Consulted: Service architect, service designer, and business analysts.

Informed: Service testers.

Service Test Results Work Product

Description: These document the results of executing the service test plan, and cover both functional and nonfunctional testing. All tests must have been successfully completed before a service can pass through the service test control point.

Purpose: Ensure the highest possible QoS.

When needed: Should be completed after testing is completed successfully and before the service acceptance control point. These test results should be examined regularly to look for any systemic issues that need to be addressed, for example, by additional training or by changing standards or best practices.

Responsible: Service testers, QA.

Accountable: Service test manager or QA.

Consulted: Service architect, service designer, and business analysts.

Informed: SOA governance lead, PMO.

Service Version Management Approach Work Product

Description: This defines how services and automated business processes will be version managed, and assigns responsibility for performing and verifying that this approach is followed in practice. It should include advice on designing services that are as forward compatible as possible—that is, that can be incrementally extended without impact to existing consumers. Chapter 6, discusses service version management in more detail.

Purpose: Define how services will be version managed.

When needed: Should be created as soon as the SOA development approach work product has been approved.

Responsible: Lead SOA architect, service architects, SOA governance lead.

Accountable: Business service champion.

Consulted: IT operations.

Informed: IT operations, service consumers.

SOA Development Performance Report Work Product

Description: These reports are based on continuously monitoring the actual analysis, design, construction, and testing efforts associated with constructing services or automating business processes.

These reports should show actual resources, explain any discrepancies with established service construction estimation metrics, and show trends. In the best case scenario, they reveal a continuous improvement in productivity and decline in rework!

Purpose: Ensure continued vitality of the SOA development approach and the vitality of SOA governance.

When needed: Should be completed and distributed every one to three months.

Responsible: Lead SOA architect, PMO, SOA governance lead.

Accountable: Lead service architect.

Consulted: Service testers.

Informed: SOA enablement team, PMO.

SOA Development Tools Work Product

Description: There are many fine software tools to assist in the development and testing of services and automated business processes, and any organization embarking on a transition to SOA should select a set of tools to improve their productivity. The most effective tools support the principle of model-driven development—that is, the creation of models of services or automated processes that are progressively refined until they are considered complete, at which time the code is generated automatically. Chapter 6 describes this in more detail.

Purpose: Create and maintain a productive development and test environment.

When needed: Should be created as soon as the SOA development approach work product has been created.

Responsible: SOA lead architect, service architects.

Accountable: SOA lead architect.

Consulted: Service developers, service assemblers, SOA business analysts, process modelers, process developers, monitoring developer, enterprise architects.

Informed: SOA enablement team.

SOA Development Lessons Learned Work Product

Description: At the end of any significant task in the service or process automation lifecycle, it is valuable to hold a relatively informal session to discuss any lessons learned from the exercise. Lessons learned should cover areas such as the SOA development process, work products, governance approach and level of governance rigor, and tools and techniques. Whether the lessons learned are positive or negative, they provide valuable insight into the vitality of the SOA development and governance process vitality.

Purpose: Ensure continued vitality of the SOA development approach and the vitality of SOA governance.

When needed: Should be completed and distributed once every four months.

Responsible: All control point reviewers.

Accountable: SOA governance lead or PMO.

Consulted: SOA enablement team.

Informed: SOA enablement team, PMO.

SOA Governance Policy Exemption Work Product

Description: The ability to grant exceptions to having to conform to common standards, best practices, or policies in case of established business need is important, and the exemptions process must be defined in the SOA governance plan, along with a definition of who is empowered to grant such exemptions. The level and type of exemptions and the reasons they were granted or refused are important measures of the SOA governance vitality:

- If there are few exemptions requested, it is possible that either the standards are too lax, the degree of governance applied is inappropriate, or that some developers are ignoring the governance approach or rules altogether.

- If there are too many exemptions requested or granted, the standards might be too restrictive.

Purpose: Record the granting or refusal of requests for exemptions from compliance with standards or best practices.

When needed: Should be completed whenever a request is received for exemption from a policy or standard.

Responsible: SOA lead architect, SOA governance lead, QA, PMO.

Accountable: Lead service architect or enterprise architect.

Consulted: Service architects.

Informed: Exemption requestor (typically a service architect or service designer).

SOA Reuse and ROI Report Work Product

Description: Based on the SOA reuse and ROI monitoring plan, this regular report should assess, as objectively as possible, the real levels of reuse and ROI that have been achieved. Some of the data to produce this report will be obtained from IT projects, through the medium of the project benefits and ROI work product.

Purpose: Measure and report the real business value of the SOA transformation initiative.

When needed: Every one to four months, or alternatively displayed on a real-time governance dashboard.

Responsible: SOA governance lead, lead service architect, PMO.

Accountable: Business service champion.

Consulted: SOA enablement team, service consumers, IT operations.

Informed: Chief executives, PMO, SOA enablement team.

The dependencies among these work products are depicted in Figure 4.3. The format is the same as for Table 3.3 in Chapter 3. Again, those work products marked with an asterisk (*) should be revised every 6 to 12 months to ensure continued vitality.

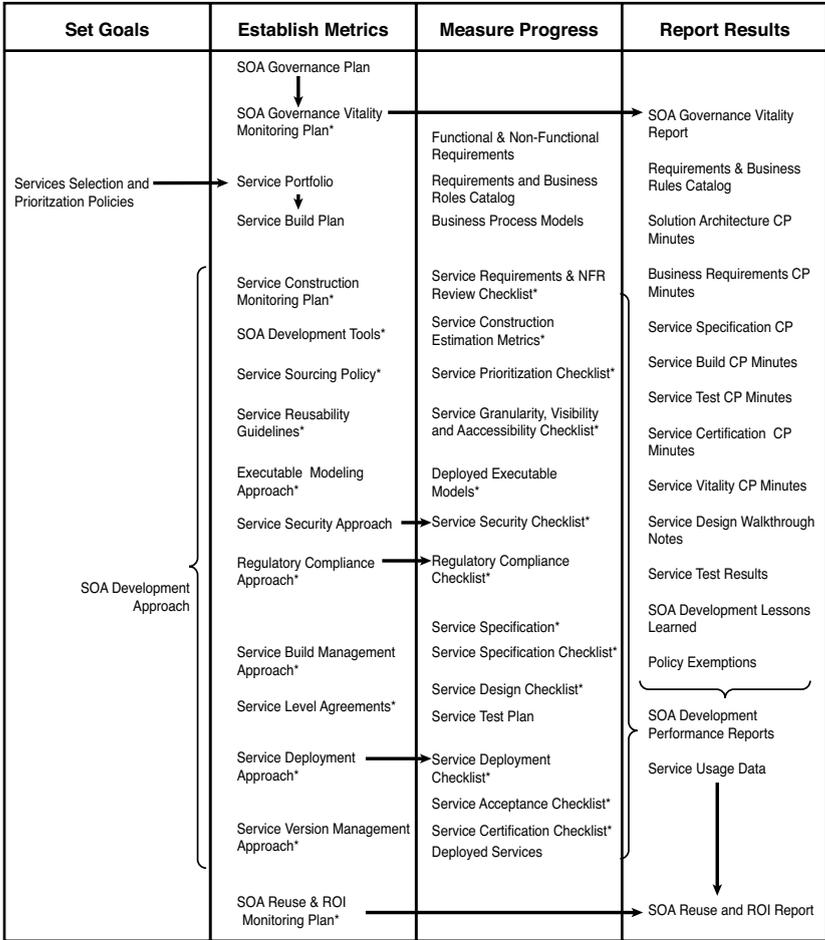


Figure 4.3 Service Development Domain: Work Product Dependencies

The Service Operations Domain

“Civilization advances by extending the number of important operations which we can perform without thinking about them”

–Alfred North Whitehead

This section covers governance of services and automated business processes in a production environment. SOA offers both benefits and challenges to IT operations. The benefits include the following:

- It’s much easier to monitor the usage and QoS provided by services, and there are some excellent commercial products that support this activity, some of which even provide real-time “dashboards” to display items such as QoS and usage statistics.
- Services offer much richer deployment opportunities and much more information about who uses them than do applications.
- A generic security mechanism can protect all services from security exposures or malicious threats.

The challenges include the following:

- SLAs need to be continuously monitored, and SLA violations need to be recorded.
- Frequent deployment of individual services can represent a threat to the stability of the production environments unless the service quality, build management, and deployment processes are highly effective.

Key SOA Governance Tasks Involved in Operating Services

Key tasks in ensuring that SOA operations are governed effectively include the following:

- Providing effective technical support for services and resolving software quality incidents efficiently and rapidly

- Monitoring the execution of services and automated business processes, most especially in terms of monitoring their compliance with the terms of their SLAs
- Maintaining the vitality, reliability, availability, and performance of the operational systems

Most organizations have a separate IT operations organization to manage the production IT systems. Table 4.2 describes the key capabilities that an IT operations organization needs to enable effective governance of services.

Table 4.2 Service Operations Domain: Capabilities, Risks, and Remedial Work Products

Capability	Associated Issues and Risks	Risk Level	Governance Work Products	Cost of Remedy
O01. Service execution monitoring	Service consumers expect high availability and good QoS—and the SLAs guarantee that they will receive them.	Critical	<ul style="list-style-type: none"> ■ SLAs ■ QoS goals ■ QoS monitoring plan ■ QoS report ■ Service usage data ■ IT capacity management plan ■ SLA compliance report ■ SLA monitoring plan 	Fairly high
O02. Service operational vitality	Service consumers do not want to have to modify their applications every time new service versions appear, unless those versions contain changes that they need.	Critical	<ul style="list-style-type: none"> ■ Service operational vitality report ■ Deprecated or decommissioned services 	Moderate
O03. Service support	Service consumers may need technical support if there are any bugs or other problems with services.	Critical	<ul style="list-style-type: none"> ■ Technical support approach and targets ■ Problem incident log 	Fairly high

Service Operations Domain Work Product Definitions

This sections contains a list of descriptions of the work products whose production can help govern the Service Operations domain. Most of these work products should exist in any moderately well-governed IT production environment, but some of these need additional extensions for SOA. Again, they are organized in alphabetic order, and work products that have already been defined in Chapter 3 are not repeated. The roles involved in creating these products were also defined in that chapter.

Deprecated or Decommissioned Services Work Product

Description: This is just a list of obsolete services that have been or will eventually be discontinued. Deprecated service should continue to be supported, but no new consumers should be able to access them. The number of such obsolete or obsolescent service versions should help determine the efficacy of the service version management approach—in an ideal world there would be few if any of these.

Purpose: Monitor SOA operational vitality.

When needed: Every four to six months.

Responsible: Service registrar

Accountable: Business service champions.

Consulted: Service consumers, PMO.

Informed: SOA enablement team, PMO, service consumers, IT operations.

Problem Incident Log Work Product

Description: This is a basic IT governance work product that should be updated to reflect the specific needs of service consumers and users of automated business processes. It records details of any technical problems that have occurred and how they were resolved.

Purpose: Used to provide invaluable information about the quality of deployed services. The SOA governance lead and lead SOA architect should look for any system quality issues and correct them as a matter of urgency.

When needed: Should be completed before any SLAs are published, and the SLAs should contain clauses that reflect the level of technical support that will be provided.

Responsible: Help desk / technical support staff.

Accountable: IT operations manager, technical support manager.

Consulted: SOA governance lead.

Informed: Problem reporter, PMO.

Quality of Service Goals Work Product

Description: QoS goals set explicit targets for performance and operational efficiency that services and automated processes will be measured against. The goals should exceed those specified in individual service nonfunctional requirements and SLAs by a small increment to ensure that those contractual conditions are more secure than the QoS goals themselves.

Purpose: Monitor performance of service operations.

When needed: Created as the first services are deployed, then updated every six months or so.

Responsible: Lead service architect, lead SOA architect, SOA governance lead.

Accountable: IT operations manager.

Consulted: Business service champions, PMO, individual business units.

Informed: IT operations, SOA enablement team, PMO.

QoS Monitoring Plan Work Product

Description: This is the plan to monitor execution of services and automated business processes against the QoS goals. This plan should include monitoring the business impact of any SOA infrastructure problem. The breakdown of a single network element might seem relatively insignificant, but if it represents a single point of failure in a major service or function of the SOA infrastructure, it may have significant business impact, unless the operational model includes full redundancy.

Purpose: Ensure that QoS goals are met.

When needed: Created as the first services are deployed, and then updated every six months or so.

Responsible: Lead service architect, lead SOA architect, SOA governance lead, monitoring developer.

Accountable: IT operations.

Consulted: Business service champions, QA, PMO.

Informed: SOA enablement team.

QoS Report Work Product

Description: This a regular report (or ideally a component on a real-time governance dashboard) that compares actual QoS with the QoS goals on a service-by-service basis.

Purpose: Ensure that QoS goals are met.

When needed: At least monthly, ideally updated online in real time.

Responsible: Lead service architect, SOA architect, SOA governance lead, monitoring developer, IT operations.

Accountable: SOA governance lead or IT operations manager.

Consulted: IT operations.

Informed: SOA enablement team, SOA executive sponsor.

Service Operational Vitality Report Work Product

Description: This is a regular summary report that provides an overview of the status of SOA operational vitality, including growth in service usage (both in terms of new service consumers and growth in service execution requests), QoS reporting, and new services deployed during each reporting period. It draws from several sources, such as the QoS report, *SLA* compliance report, and deployed services and service usage data provided by IT operations.

Purpose: Communicate SOA operational vitality.

Responsible: SOA governance lead, monitoring developer, and PMO define the structure and layout of the report; IT operations and service registrar provide the data.

Accountable: IT operations manager or SOA governance lead.

Consulted: Business service champions, IT operations.

Informed: SOA enablement team, SOA executive sponsor.

Service Usage Data Work Product

Description: One of the major advantages of SOA over more traditional software development styles is that it is possible to capture a great deal of operational information about usage of services and automated processes. In fact, it is impossible to govern the Service Operations domain, or to bill third parties for service usage, unless this information is captured and recorded.

It is impossible to capture usage statistics manually, so it is essential that the SOA infrastructure itself automates this task. Data that should be captured for both services and automated processes include the following:

- Who invoked each service operation or automated process?
- Were there any attempts to invoke a service or process by unauthorized requesters?
- What was the total time taken to execute the request, and was it within the SLA and QoS targets?
- What were the overall transaction rates for each service, hour by hour?

In the case of automated processes, additional data needs to be recorded:

- Which of the several possible logical branches were taken in each execution?
- When an automated process invoked a manual task, how long before that task completed?
- Are any manual tasks “hung up”—that is, not completed within the QoS thresholds for that task?

The volume of data this represents will be formidable, and it will need to be summarized in a report or dashboard display.

Purpose: Capture and summarize essential data on performance and utilization of services and automated processes.

Responsible: SOA governance lead, monitoring developer, and PMO define the structure and layout of the report or dashboard display; IT operations provides the raw data.

Accountable: IT operations manager or SOA governance lead.

Consulted: Business service champions, IT operations.

Informed: SOA enablement team, SOA executive sponsor.

SLA Compliance Report Work Product

Description: There are two types of SLA compliance report:

- Regular SLA compliance confirmation. Most, and ideally all, SLA compliance monitoring reports should show actual QoS numbers achieved that are well within the SLA terms.

- The more important—but rare, we hope—SLA compliance failure reports. In the event of SLA compliance failures, notifications should be sent urgently to the service consumers, the lead SOA architect, business service champions, and the SOA governance lead, apprising them of the situation and the actions taken to prevent recurrence.

Purpose: Monitor SLA compliance and handle incidents.

When needed: Regular SLA compliance reporting should be performed monthly, in support of the SLA monitoring plan described next. In the event of a SLA violation, immediate corrective action should be taken, and all effected service consumers informed as soon as possible.

Responsible: IT operations, monitoring developer, SOA architect, service architects.

Accountable: IT operations manager.

Consulted: Service architect, service designer, QA in the event of SLA violations.

Informed: PMO, SOA enablement team, help desk / technical support staff, service consumers.

SLA Monitoring Plan Work Product

Description: This is the plan to monitor all service and process executions, consumer by consumer, to ensure that they are executed with the terms of the SLAs appropriate to that individual consumer. SLA monitoring needs to be performed using a formal systems management tool or framework; it is not a task that can be performed manually.

Purpose: There is no point in having SLAs if you don't know whether you are meeting them. There is little value in SOA governance without formal SLAs being in place and enforced.

When needed: At the same time as the SLAs are defined.

Responsible: PMO, lead service architect, lead SOA architect, IT operations, SOA governance lead, monitoring developer.

Accountable: SOA governance lead or PMO.

Consulted: Business service champions, IT operations.

Informed: IT operations, QA, PMO, SOA enablement team.

Technical Support Approach and Targets Work Product

Description: This is a basic IT governance work product that should be updated to reflect the specific needs of service consumers and users of automated business processes. It defines the level and speed of technical

support that internal and external service consumers can expect. Enhanced levels of support should be provided for high-severity problems, and for high-value services. Assured levels of technical support should be included in service and automated process SLAs.

Purpose: Ensure that and SLAs honored and that service consumers receive adequate technical support.

When needed: Should be completed before any SLAs are published, and the SLAs should contain clauses that reflect the level of technical support that will be provided.

Responsible: Lead service architect, SOA governance lead, PMO.

Accountable: IT operations manager.

Consulted: Business service champions.

Informed: Help desk / technical support manager, QA, PMO.

The dependencies among work products are shown in Figure 4.4. Again, the format is the same as for Figure 3.3 (in Chapter 3) and Figure 4.3.

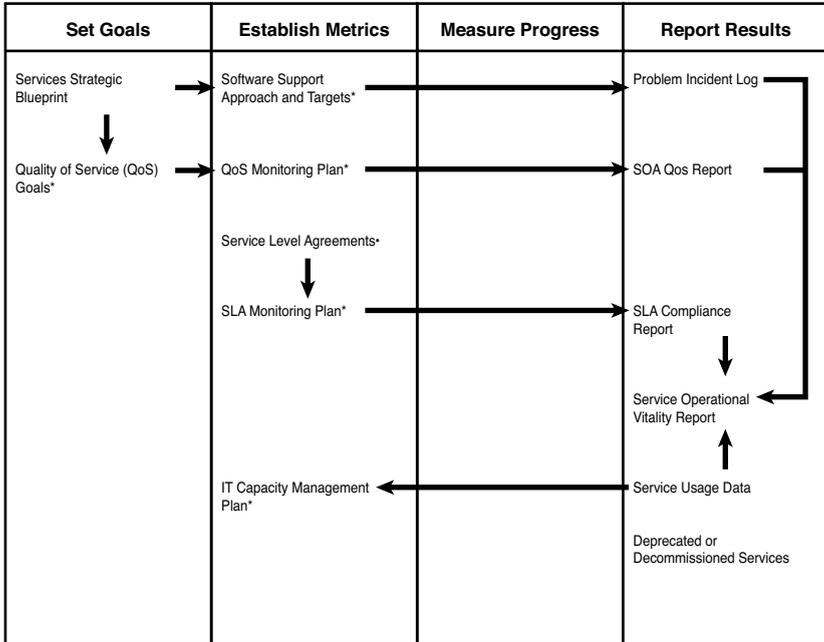


Figure 4.4 Service Operation Domain: Work Product Dependencies

Case Study

“To regret one’s own experiences is to arrest one’s own development. To deny one’s own experiences is to put a lie into the lips of one’s life. It is no less than a denial of the soul.”

—Oscar Wilde

Service transformation planning was one of the capability priorities that SOA governance needed so that it could work with the EA group and help govern the resultant business service priorities. Service development lifecycle controls (SDLCs) represent another area of great concern. After all, the adherence to best practices and a consistency of approach across the development organizations are needed, especially to help enforce the results of the service transformation planning.

Service Development Lifecycle Controls

Ideation has known for some time that it needs a common and structured development process. Some groups perform peer reviews from time to time on their code, but the reviews tend to be haphazardly completed.

Another group at Ideation is considering a standardized development approach. Your SOA governance team has been asked to propose governance control gates for the service development lifecycle of

- Business requirements
- Solution architecture
- Service identification and specification
- Service build
- Service test
- Service certification

Control Gates for Ideation

In consultation with the development stakeholders, you decide to create the following control gates for the Ideation SDLC:

- Business Requirements control gate
- Solution Architecture control gate
- Service Design control gate
- Service Build control gate
- Service Test control gate
- Service Certification and Deployment control gate

As a first step, you create the template for the Business Requirements control gate and elicit feedback, as shown in Table 4.3.

Table 4.3 Ideation Business Requirements Control Gate

Section	Contains	Definition
Motivation	The justification for this control gate.	We must have a consistent and well-formed set of business requirements to create services that give us the greatest amount of agility and reuse. This requires that we have a consistent and repeatable approach to the specification of requirements and that this specification gives the required information for the rest of the development lifecycle.
Objective	The output objective	Ensure that the business requirements have a sufficient level of content regardless of form so that development can deliver on the requirement; IT will have a true specification of the business need and objective. This better enables IT to size and estimate this project. Enables the alignment of cross-functional teams on the delivery of the business requirement.
Trigger	Triggering event	Complete business requirements are delivered by the business to the project manager.

Section	Contains	Definition
Scale / Applicability Guidance	Indication of what scale/style of review is appropriate for the circumstance	Perform if project is > \$100K in size.
Review type	<p>General classification of review types:</p> <ul style="list-style-type: none"> ■ Peer review ■ Submission to approver ■ Workshop ■ Formal meeting 	<p>A workshop to review the requirements artifact and identify that the correct level of detail per the executive design authority standard for business requirements specification has been followed. The following roles are represented at the workshop:</p> <p>Application architect Process analyst Business architecture Business analyst Requirements analyst Solution designer Project manager</p>
Concept/approach	Description of how the review is conducted and how it supports the objective	<p>The project manager schedules and leads the workshop.</p> <p>The application architect, business analyst, requirements analyst, and solution designer must validate the requirements while meeting the needs of the next steps of the development lifecycle. That is, they have the information they need to do a good job and the requirements are understood.</p> <p>The business architecture must validate the requirements by following the architecture review board (ARB) requirements standard. Needed updates to requirements are noted and assigned for follow up, or requirements are approved or disapproved (with updates required).</p>
	Governing authority	Business architecture is the governing authority responsible for validating form and that all standards for business requirements are followed.
	Responsible/manages	The project manager schedules and leads the workshop.
	Accountable - approves/conditional approval/rejects	The head of line of business (LoB) must be accountable and approve the business requirements with a final signoff.

Table 4.3 Ideation Business Requirements Control Gate (continued)

Section	Contains	Definition
	Consults/supports/performs	<p>The application architect understands the requirements and ensures all the requirements information needed for the high-level solution architecture (the architecture solution document) is there. The process analyst answers queries about the business process and ensures that the rest of the requirements conform to the applicable processes.</p> <p>The business analyst is responsible for delivering the requirements and is available to answer questions about the requirements or to follow up with any updates needed.</p> <p>The requirements analyst understands the requirements, creates the specification, and identifies the specifications applicable to each development group (to understand which areas are impacted).</p> <p>The solution designer understands the requirements and will take the architecture solution document from the application architect and create the solution design (end-to-end solution). The solution designer also oversees the components of the solution design.</p> <p>Test/QA—Is this requirement sufficiently articulated to be validated?</p>
	Informed	All participants, governance specialist.
Artifacts to be made available	The input that is required	<p>Business requirements should be circulated before the workshop to all participants. (A customer service level agreement [SLA] needs to be considered and agreed to.)</p> <p>Opinions must be sought from the application architect, business analyst, requirements analyst, and solution designer to validate the requirements.</p>

Section	Contains	Definition
	What opinions to be sought	The application architect, business analyst, requirements analyst, and solution designer must validate that the requirements meet the needs of the next steps of the development lifecycle. That is, they have the information they need to do a good job, and the requirements are understood.
Key review criteria	Highlights the key criteria	The business requirements standard must be adhered to, including the following: <ul style="list-style-type: none"> ■ Consistency of requirements within the business requirements package ■ Adherence of business requirements to business processes ■ Adherence of business requirements to regulatory and organization compliances and standards
Key work product acceptance criteria	Key acceptance criteria to be met for successful review of work product	Consistency and completeness of the business requirements must be followed. Any inconsistencies or gaps or opportunities for improvement or rework of the business requirements must be specified. Record next steps, work assigned, and schedule agreed. Any follow up agreed.
Checklist	Sets a suggested process for the review. Provides objective support for recommendations, rework requests, and signoffs.	Business requirements checklist will be used.
Escalation process	Name of escalation process to be used to request an exception to the control gate result.	Escalates a decision to the program management office (PMO) to drive the business requirements process to conclusion.
Measurements	Metrics to be captured during or at the end of this control gate.	Governance metrics on this gate, including incrementing the number of times this gate has been implemented, incrementing the number of times this business unit has been through this gate, and the results (pass, fail, pass with conditions), and the checklist score.
Outcomes	Who is responsible for final signoff, who gets notified, and who are the results delivered to?	Final signoff—Head of the LoB. Notification—All on the informed list and service registrar. Delivery—The PMO will be the official owner of the results.

As a next step, you need to create and get feedback on the control gate for the Solution Architecture control gate, as shown in Table 4.4.

Table 4.4 Ideation Solution Architecture Control Gate

Section	Contains	Definition
Name	Control gate name	Solution Architecture
Motivation	The justification for this control gate	Approval to proceed with proposed solutions presented in conformance to technology standards and principles: <ul style="list-style-type: none"> ■ To confirm deliverables are encompassing and “fit for purpose” ■ To promote deliverable integration and support ■ To proactively understand impacts before further investment
Objective	The output objective	To assess deliverable’s impact on member domains and resulting consideration: <ul style="list-style-type: none"> ■ To agree on integration, interfaces, and support of deliverables ■ To ensure alternative approaches and solutions ■ To approve new IT solutions presented.
Trigger	Triggering event	Project manager receives completed solution architecture document.
Scale / applicability guidance	Indication of what scale/ style of review is appropriate for the circumstance	Perform if project size > \$100K.
Review type	General classification of review types: <ul style="list-style-type: none"> ■ Peer review ■ Submission to approver ■ Workshop ■ Formal meeting 	A workshop to review the requirements artifact and identify that the correct level of detail per the guidelines has been specified. The following roles are needed: <ul style="list-style-type: none"> Application architect Process analyst Business architecture Business analyst Requirements analyst Solution designer Head of line of business

Section	Contains	Definition
Concept/approach	Description of how the review is conducted and how it supports the objective	The business analyst must validate the requirements as meeting the standards for requirements, and must then trigger the business requirement review workshop to be scheduled and completed.
Participants, roles, and interests	Governing authority	ARB.
	Responsible/manages	Project manager—Trigger the meeting and provide follow up and results.
	Accountable - approves/conditional approval/rejects	Lead architect—Responsible for presenting and answering question on the architecture solution document.
	Consults/supports/performs	<p>Solution designer—Responsible for representing the technical expertise of standards and policies and evaluating the architecture solution document end-to-end solution as being feasible.</p> <p>Business analyst—Responsible for representing the business client and validates that the solution meets needs of the business and accepts or declines proposed trade-offs.</p> <p>Architecture executives—Provide guidance, review output, and ensure governance.</p> <p>Infrastructure architect (if heavy infrastructure in solution)—Have provided infrastructure architecture in the architecture solution document and answer questions as needed.</p> <p>Infrastructure designer—Validate the solution architecture for structural feasibility.</p> <p>Requirements analyst—Review and understand the high-level solution architecture and context of downstream requirements and analyst comments.</p> <p>Data architect—Provided data architecture in the architecture solution document and answers questions as needed.</p>
	Informed	All participants, Center of Excellence (CoE).

Table 4.4 Ideation Solution Architecture Control Gate (continued)

Section	Contains	Definition
Artifacts to be made	The input that is required	Architecture solution document available.
	What opinions to be sought	The solution designer, solution architect, infrastructure designer, and the requirements analyst must validate the architecture solution document as meeting the needs of the next steps of the development lifecycle. That is, they have the information they need to do a good job, and the high-level design is understood.
Key review criteria	Highlights the key criteria	The architecture solution document must comply with architectural standards, policies, and patterns. The architecture solution document conforms to existing application future views where those views exist. The architecture solution document highlights key infrastructure impacts and requirements. Have services been identified and classified, and are they at the correct level of granularity (services litmus tests)?
Key work product acceptance criteria	Key acceptance criteria to be met for successful review of work product	Consistency and completeness of the high-level design must be followed. Any inconsistencies or gaps or opportunities for improvement or rework of the high-level design must be specified. Record next steps, work assigned, and schedule agreed. Any follow up agreed.
Checklist	Sets a suggested process for the review. Provides objective support for recommendations, rework requests, and signoffs.	Follow the architecture solution document checklist.
Escalation process	Name of escalation process to be used to request an exception to the control gate result	Exceptions to demands for rework can be approved by the ARB. Where the issue is a dispute on service identification, the ARB shall send that issue to the CoE for resolution.

Section	Contains	Definition
Measurements	Metrics to be captured during or at the end of this control gate	Governance metrics on this gate, including incrementing the number of times this gate has been implemented, incrementing the number of times this lead architect has been through this gate, and the results (pass, fail, pass with conditions), and the checklist score
Outcomes	Who is responsible for final signoff, who gets notified, and who are the results delivered to?	Final signoff—ARB lead. Notification—All participants are notified and service registrar. Handoff—The PMO will be the official owner of the results.

Conclusion

“Every kind of peaceful cooperation among men is primarily based on mutual trust and only secondarily on institutions such as courts of justice and police”

—Albert Einstein

This chapter described a practical work-product approach to governing an efficient service factory—a software engineering-based approach to defining, developing, testing, deploying, and operating functional services and automated business processes.

The work-product approach helps to manage the complexity and interdependencies of the tasks involved, by introducing several intermediate deliverables, documenting the transitions between the various states of service and project lifecycles, and creating a set of control points to act as a governance mechanism to ensure that those deliverables have been constructed to the quality needed. Governance is enabled by formal or informal contracts between work-product creators and work-product consumers; such an arrangement enables early detection and resolution of any quality issues, instead of relying on the imposition of rigid centralized controls.

Our approach is not authoritarian—such an approach simply doesn't work in managing anything nearly as complex as migration to SOA. Instead, our approach uses work products and short, efficient control point reviews to act as a collaborative set of checks and balances to apply continuous feedback to improve the quality and completeness of the SOA assets that the enterprise deploys.

In Chapter 6, we revisit the service factory to look a little more closely at the how the production line operates, and how governance is applied to individual transitions in the life of each service or automated process.

Index

A

- Adams, Scott, 267
- adopting SOA (service-oriented architecture), 4-8
 - Capability Assessment Tool (CAT), 7
 - communication, 7
 - estimating SOA projects, 8
 - organizational maturity, 7
 - scope of SOA transformation, 5
 - SOA Center of Excellence (CoE), 8
 - standardization, 4
- alternatives, evaluating, 231
 - governance mechanism recommendations, 234-235
 - process identification, 232-234
- Amiel, Henri, 68
- analyzing assessment results, 231
- Anarchy model, 70
- application portfolio rationalization, 316
- architecture standards and design
 - patterns work product, 122-123
- assess governance capability phase, 323-324
- assessing organizations
 - governance priorities and near-term goals, 73-83
 - Ideation Communications case study, 84-88, 334-337
 - organization type, 69-70
 - suitability considerations, 70

- symptoms and causes, 60-68
- understanding history of, 57-60
- automated process lifecycle, 287
- automating tasks, 294-295

B

- Baglivi, Giorgio, 84, 333
- benefits
 - of production engineering approach, 95-97
 - of SOA (service-oriented architecture), 2-4
- bibliography, 371-372
- bottom-up approach to defining goals, 313-317
- bottom-up requirements analysis, 270-271
- Brandeis, Louis, 260, 348
- business agility monitoring plan work product, 123
- business agility monitoring report work product, 152
- business analysts, 102-104
- business and IT financial priorities
 - work product, 125
- business discovery, 227
 - defining SOA vision and strategy, 228
 - determining existing governance structure, 229
 - identifying SOA business principles, 228

business flexibility monitoring plan
work product, 124

business flexibility monitoring report
work product, 153

business “heat map” work product,
124-125

Business Monarchy, 69

business principles
identifying, 228
updating, 236-237

business requirements and service
identification control point, 172

business rules and requirements work
product, 125-126

business service champions, 101-102

business term glossary work
product, 126

business to IT alignment, 314

Business Vision & IT Alignment
capability, 52

business vision work product, 126

C

Caldwell, Philip, 43

Capability Assessment Tool (CAT), 7

capturing requirements
bottom up, 270-271
top down, 267-270

Carroll, Lewis, 309

case study. *See* Ideation

Communications case study

CAT (Capability Assessment Tool), 7

Center of Excellence. *See* CoE

certification
Ideation Communications case study,
304-305
certification detail, 355-356
certification header, 355
overview, 353-355
service certification detail, 306-307
service certification header, 306

preparing for, 283-284

service certification and deployment
point, 176-177

Change Management capability, 54

characteristics of SOA governance,
43-46

Churchill, Winston, 281, 353

CIM (Common Information Model), 4

COBIT (Control Objectives for
Information and Related
Technology), 24, 60

CoE (Center of Excellence), 8
creating, 237-240
mission statements, 349

collection, requirements collection,
166-167

Common Information Model (CIM), 4

common pitfalls of SOA governance,
46-50

communication, 7
communication plans, 254-255
communication process, 30
with service customers, 303
skip-level communication, 60

companies, assessing
governance priorities and near-term
goals, 73-83

Ideation Communications case study,
84-88, 334-337
organization type, 69-70
suitability considerations, 70
symptoms and causes, 60-68
understanding history of, 57-60

complexity management, 71-72, 91-94

compliance
goals and measurements, 315
process, 28

conducting
kickoff meetings, 227
planning workshops, 226

Control Objectives for Information
and Related Technology (COBIT),
24, 60

control point minutes, 184

control points
business requirements and service
identification, 172

Ideation Communications case study
Business Requirements control gate,
 341-345
Solution Architecture control gate, 345-348

requirements capture and
 service identification
capturing requirements bottom up, 270-271
capturing requirements top down, 267-270
triage process, 271-277

service build, 175

service certification and deployment
overview, 176-177
preparing for, 283-284

service design
overview, 174-175
preparing for, 279-280

service realization, 281-282

service specification
overview, 174
preparing for, 278-279

service test
overview, 175-176
preparing for, 282-283

service vitality
overview, 177
preparing for, 285

solution architecture
overview, 173
preparing for, 277-278

Cook, Rick, 165

core competencies
 operational management and service
 monitoring, 168
 requirements collection, 166-167
 service design, 167
 service development, 167
 service testing and deployment, 168

corporate governance, 14-15

current state documentation, gathering,
 225-226

customers
 communicating with, 303
 customer intimacy, 312-313

D

define phase of SOA
 implementation, 235

CoE, creating, 237-240

governance processes, modifying/defining,
 240-241

illustration, 236

SOA governance capability gaps, closing,
 242-245

SOA governance infrastructure and tools,
 245-247

SOA governance plans, 247-249

SOA principles, refining, 236-237

dependencies
 in SOA Plan & Organize domain,
 144-146
 in SOA Program Management Controls
 domain, 160

deployment, 168
 preparing for, 283-284
 service certification and deployment
 point, 176-177

deprecated or decommissioned services
 work product, 203

design, service design control point,
 167, 174-175

develop governance enhancements
 phase, 324

The Discipline of Market Leaders (Treacy
 and Wiersema), 310

discrete operations, 299

documentation on current state,
 gathering, 225-226

Douglas, Frederick, 249

Drucker, Peter F., 46

E

Edison, Thomas, 321

EDM (enterprise data model) work
 product, 127-128

education plans, 253

Einstein, Albert, 217

enable phase of SOA
 implementation, 249
 communication plans, 254-255
 education plans, 253
 executing measurement, 257-259
 illustration, 250, 256-257
 incorporating governance mechanisms
 with CoE, 251-253
 mentoring plans, 254
 tools and infrastructure, 255-256
 transition plans, 250-251
 enterprise data model (EDM) work
 product, 127-128
 enterprise governance, 15
 Enterprise Program Management
 capability, 54
 ERB (executive review board), 311
 Ideation Communications case study, 350
 estimating SOA projects, 8
 evaluate management performance
 phase, 322-323
 Event Management & Service
 Monitoring process, 32
 Exception/Escalation process, 33
 exceptions and appeals process, 29
 executable modeling approach work
 product, 185
 executing measurement, 257-259
 executive review board. *See* ERB
 executive sponsors, 100
 existing governance structure,
 determining, 229

F

Federal model, 70
 feedback process, 321-324
 assess governance capability phase,
 323-324
 develop governance enhancements
 phase, 324
 evaluate management performance
 phase, 322-323
 Feudal model, 69
 Fischer, Martin H., 50, 57
 Ford, Henry, 89, 94

Franklin, Benjamin, 282, 301
 Fried, Jason, 307
 functional and nonfunctional checklist
 work product, 186
 functional and nonfunctional
 requirements work product,
 185-186
 functional work products, 92
 funding, Ideation Communications case
 study, 263-264, 351-353
 future-proof services, designing,
 302-303

G

Ghandi, Mahatma, 4
 glossary, 361-369
 goals and measurements
 application portfolio rationalization, 316
 assessing goals, 73-80, 82-83
 business to IT alignment, 314
 customer intimacy, 312
 defining with bottom-up approach,
 313-317
 defining with top-down approach,
 311-313
 governance process compliance, 315
 governance process maturity, 316
 maturity index, 315
 overview, 310
 product leadership, 312
 reference architecture, 314
 ROI (return on investment), 314
 service broker usage, 317
 service identification technique, 315
 service response time, 317
 service reuse, 313
 TCO (total cost of ownership), 313
 Goethe, Johann Wolfgang von, 356
 Goldwyn, Samuel, 278
 governance
 capability gaps, closing, 242-245
 corporate governance, 14-15
 definition of, 12-14
 enterprise governance, 15

governance processes
communications, 30
compliance, 28
exceptions and appeals, 29
illustration, 27
vitality, 28

IT governance
overview, 15-16
reference sources, 22-25

mechanisms
infrastructure and tools, 35-36
monitors and metrics, 34
organizational change management, 35
principles, policies, standards, and procedures, 33-34
recommendations, 234-235
skills, 35

SOA governance. *See* SOA governance

governance paradigm (SOA), 18-22

governance-related work products, 92

governed processes, 25, 30-33

granularity, regulating
 automation, 294-295
 discrete, standalone operations, 299
 overview, 287-288
 read-only services, 300
 reusability of operations, 295-297
 scope of operations, 297-299
 tasks versus processes, 288-293
 update services, 300-301

H-I

Hippocrates, 36, 330
 Hughes, Langston, 18

IBM Process Reference Model for IT (PRM-IT), 16

Ideation Communications case study
 background, 330-333
 business goals, 38-40, 333
 Center of Excellence (CoE) mission statement, 349
 company background, 36-38

control gates
Business Requirements control gate, 341-345
Solution Architecture control gate, 345-348

current state of, 332

executive review board (ERB) mission statement, 350

mission statement, 260-262

organizational diagram, 262, 350-351

overview, 36

SDLC (system development lifecycle)
business requirements control gate, 210-213
service development lifecycle controls, 209
solution architecture control gate, 214-217

service certification, 304-305
certification detail, 355-356
certification header, 355
overview, 353-355
service certification detail, 306-307
service certification header, 306

service development lifecycle controls, 340

service governance vitality, 325-328, 357-359

service transformation planning, 161-163

service ownership and funding, 263-264, 351-353

service transformation planning, 337-340

SOA planning assessment, 84-88, 334-337

systems and networks, 331

Identify and Allocate Costs
 capability, 54

identifying
 governance processes, 232-234
 services. *See* service identification

implementation (of SOA governance model)
 define phase, 235
CoE, creating, 237-240
governance processes, modifying/defining, 240-241
illustration, 236
SOA governance capability gaps, closing, 242-245

SOA governance infrastructure and tools,
defining, 245-247

SOA governance plans, *defining*, 247-249

SOA principles, *refining*, 236-237

enable phase, 249

communication plans, 254-255

education plans, 253

illustration, 250

incorporating governance mechanisms with CoE, 251-253

mentoring plans, 254

tools and infrastructure, 255-256

transition plans, 250-251

measure phase

executing measurement, 257-259

illustration, 256-257

overview, 219-220

plan phase, 223

alternatives and phase recommendations, 231-235

illustration, 224

IT governance environment readiness, 229-231

project startup, 224-227

SOA business discovery, 227-229

SOA governance staffing, 221-222

timeline, 223

information systems strategy blueprint
 work product, 128

Information Technology Information
 Library (ITIL), 23

Information Transformation Planning
 capability, 52

infrastructure, 35-36

defining, 245-247

implementing, 255-256

ISACF (IT Governance Institute and the
 Information Systems Audit and
 Control Foundation), 24

IT capacity management plan work
 product, 129

IT Duopoly, 70

IT governance

environment readiness, 229

analyzing assessment results, 231

*conducting governance planning
 assessment*, 230

*tailoring governance planning
 assessment*, 230

overview, 15-16

reference sources, 22

ITGI (IT Governance Institute), 24-25

ITIL (Information Technology Information
 Library), 23

IT Governance (Weill and Ross), 69

IT Governance Institute (ITGI),
 24-25, 60

IT Governance Institute and the
 Information Systems Audit and
 Control Foundation (ISACF), 24

IT Monarchy, 69

ITGI (IT Governance Institute),
 24-25, 60

ITIL (Information Technology
 Information Library), 23

J-K

key governance considerations, 20-21

key governance decisions, 21

kickoff meetings, 227

Kozol, Jonathan, 73

L

Lakein, Alan, 161, 337

lead service architects, 106-107

Lee, Spike, 219

LePatner, Barry, 277

lifecycle management. *See also* SDLC
 (system development lifecycle)

automated process lifecycle, 287

Ideation Communications case study,
 304-305, 340

service certification, 353-356

service certification detail, 306-307

service certification header, 306

overview, 265-267
 regulation of service granularity
automation, 294-295
discrete, standalone operations, 299
overview, 287-288
read-only services, 300
reusability of operations, 295-297
scope of operations, 297-299
tasks versus processes, 288-293
update services, 300-301
 requirements capture
capturing requirements bottom up, 270-271
capturing requirements top down, 267-270
 service certification and deployment,
 283-284
 service design, 279-280
 service identification, 271-277
 service realization, 281-282
 service specification, 278-279
 service testing, 282-283
 service versioning management, 301
communicating with service customers, 303
designing future-proof services, 302-303
 service vitality, 285
 services lifecycle, 285-286
 solution architecture, 277-278
 transitions, 266
 Lincoln, Abraham, 220, 310

M

Machiavelli, Niccolo, 6
 Manage Services process, 33
 Manage the Service Investment
 capability, 52
 Mandela, Nelson, 12, 40
 maturity index, 315
 maturity levels, 61-62
 measurements. *See* metrics
 mechanisms of governance
 infrastructure and tools, 35-36
 monitors and metrics, 34
 organizational change management, 35
 principles, policies, standards, and
 procedures, 33-34
 skills, 35

meetings, kickoff meetings, 227
 mentoring plans, 254
 messaging model (MM) work product,
 127-128
 metrics
 application portfolio rationalization, 316
 business to IT alignment, 314
 customer intimacy, 312
 defining with bottom-up approach,
 313-317
 defining with top-down approach,
 311-313
 executing, 257-259
 governance process compliance, 315
 governance process maturity, 316
 Ideation Communications case study,
 357-359
 maturity index, 315
 overview, 34, 310
 product leadership, 312
 reference architecture, 314
 ROI (return on investment), 314
 service broker usage, 317
 service identification technique, 315
 service response time, 317
 service reuse, 313
 TCO (total cost of ownership), 313
 mission statements (Ideation
 Communications case study),
 260-262
 Center of Excellence (CoE) mission
 statement, 349
 executive review board (ERB) mission
 statement, 350
 MM (messaging model) work
 product, 127-128
 modifying governance processes,
 240-241
 Monitor Business Benefits of SOA
 capability, 54
 monitoring developers, 112-113
 monitoring services, 168
 monitors, 34
 Moon, Elizabeth, 146

N-O

operational excellence, defining goals for, 311-312

operational management, 168

operational model work product, 129-130

operations, 273

organizational change management, 35

organizational diagram (Ideation Communications case study), 262, 350-351

organizational maturity, 7

organizations, assessing

- governance priorities and near-term goals, 73-83
- history, 57-60
- Ideation Communications case study, 84-88
- organization type, 69-70
- suitability considerations, 70
- symptoms and causes, 60-68

ownership (Ideation Communications case study), 263-264, 351-353

P

Pascal, Blaise, 318

pattern for process improvement, 90-91

- complexity management in, 71-72, 91-94
- roles and responsibilities in, 97-113
- service factory creation, 94-97

phase recommendations,

- developing, 231
- governance mechanism recommendations, 234-235
- process identification, 232-234

Plan & Organize domain (SOA), 50-53, 114-122

- architecture standards and design patterns work product, 122-123
- business agility monitoring plan work product, 123
- business and IT financial priorities work product, 125
- business flexibility monitoring plan work product, 124
- business "heat map" work product, 124-125
- business rules and requirements work product, 125-126
- business term glossary work product, 126
- business vision work product, 126
- dependencies in, 144-146
- enterprise data model (EDM) work product, 127-128
- information systems strategy blueprint work product, 128
- IT capacity management plan work product, 129
- messaging model (MM) work product, 127-128
- operational model work product, 129-130
- service portfolio work product, 131-132
- service prioritization checklist work product, 132-133
- services ownership and funding models work product, 130-131
- services selection and prioritization policies work product, 133-134
- services strategic blueprint work product, 134-135
- SOA career paths work product, 135
- SOA communications plan work product, 135-137
- SOA education plan work product, 137
- SOA governance plan work product, 137-139
- SOA governance vitality monitoring plan work product, 139-140
- SOA governance vitality report work product, 140-141
- SOA reuse and ROI monitoring plan work product, 141-142
- SOA roles and responsibilities work product, 142-143
- strategic business plan work product, 143
- supply chain and procurement processes for SOA work product, 143
- technology blueprint work product, 144

- plan phase of SOA implementation, 223
 - alternatives and phase
 - recommendations, 231
 - governance mechanism recommendations*, 234-235
 - process identification*, 232-234
 - illustration, 224
 - IT governance environment
 - readiness, 229
 - analyzing assessment results*, 231
 - conducting governance planning assessment*, 230
 - tailoring governance planning assessment*, 230
 - project startup, 224
 - conducting kickoff meetings*, 227
 - conducting planning workshops*, 226
 - gathering current state documentation*, 225-226
 - SOA business discovery, 227
 - defining SOA vision and strategy*, 228
 - determining existing governance structure*, 229
 - identifying SOA business principles*, 228
- planning assessment. *See*
 - assessing organizations
- planning workshops, conducting, 226
- plans
 - communication plans, 254-255
 - defining, 247-249
 - education plans, 253
 - mentoring plans, 254
 - transition plans, 250-251
- policies, 33-34
- Powell, Colin, 22, 33, 223
- Preston, John, 42
- principles, 33-34
- priorities, assessing, 73-83
- PRM-IT (IBM Process Reference Model for IT), 16
- problem incident log work product, 203-204
- problems with SOA governance, 46-50
- procedures, 33-34
- process developers, 111-112
- process improvement patterns, 90-91
 - complexity management in, 71-72, 91-94
 - roles and responsibilities in, 97-113
 - service factory creation, 94-97
- process modelers, 110-111
- processes, 30-33
 - compliance, 28, 315
 - exceptions and appeals, 29
 - governed processes, 25, 30-33
 - governance processes, modifying/defining, 240-241
 - identifying, 232-234
 - improvement patterns, 90-91
 - complexity management in*, 71-72, 91-94
 - roles and responsibilities in*, 97-113
 - service factory creation*, 94-97
 - Manage Services, 33
 - maturity, 316
 - processes to be governed, 30-33
 - versus tasks, 288-293
 - vitality, 28
- Procurement of Resources capability, 54
- product designers, 95
- product leadership, defining goals
 - for, 312
- production engineering, applying to
 - SOA governance, 94-97
- production engineers, 95
- production managers, 95
- Program Management Controls domain, 53-54, 146-150
 - business agility monitoring report work product, 152
 - business flexibility monitoring report work product, 153
 - dependencies in, 160
 - project actual costs work product, 153
 - project audit findings work product, 153
 - project benefits and ROI monitoring plan work product, 154
 - project benefits and ROI report work product, 154
 - project change request work product, 155
 - project context and scope work product, 155

project control point minutes work product, 155
 project dependencies work product, 156
 project deployment control point checklist work product, 156
 project execution monitoring plan work product, 156
 project financial approval control point checklist work product, 157
 project lessons learned work product, 157
 project outline approval control point checklist work product, 158
 project requirements and scope control point checklist work product, 158
 project status report work product, 158
 project workloads estimate work product, 159
 vendor assessment guidelines for SOA work product, 159
 vendor guidance notes for SOA work product, 160
project actual costs work product, 153
project audit findings work product, 153
project benefits and ROI monitoring plan work product, 154
project benefits and ROI report work product, 154
project change request work product, 155
project context and scope work product, 155
project control point minutes work product, 155
project dependencies work product, 156
project deployment control point checklist work product, 156
project execution monitoring plan work product, 156
project financial approval control point checklist work product, 157
project lessons learned work product, 157

project outline approval control point checklist work product, 158
project requirements and scope control point checklist work product, 158
project startup, 224
 conducting kickoff meetings, 227
 conducting planning workshops, 226
 gathering current state documentation, 225-226
project status report work product, 158
project workloads estimate work product, 159

Q-R

QoS (quality of service)
 quality assurance levels for work products, 71-72
 quality of service goals work product, 204
 quality of service monitoring plan work product, 204
 quality of service report work product, 205
RACI (responsible, accountable, consulted, and informed) model (project management), 93
read-only services, 300
Reagan, Ronald, 283
reference architecture, 314
reference sources for IT governance, 22
 ITGI (IT Governance Institute), 24-25
 ITIL (Information Technology Information Library), 23
refining SOA principles, 236-237
regulating service granularity
 automation, 294-295
 discrete, standalone operations, 299
 overview, 287-288
 read-only services, 300
 reusability of operations, 295-297
 scope of operations, 297-299
 tasks versus processes, 288-293
 update services, 300-301

regulatory compliance approach work product, 186

regulatory compliance checklist work product, 187

Reid, Chuck, 163

reports, 318-321

- quality of service report work product, 205
- service operational vitality report work product, 205
- SLA compliance report work product, 206-207
- SOA development performance report work product, 197

Requirements Gathering & Prioritization capability, 55

- capturing requirements bottom up, 270-271
- capturing requirements top down, 267-270
- requirements collection, 166-167
- triage process, 271-277

response time, 317

responsibilities in service factories, 97-113

responsible, accountable, consulted, and informed (RACI) model, 93

return on investment (ROI), 314

reusability of operations, 295-297

risks in SOA adoption, 4-8

- Capability Assessment Tool (CAT), 7
- communication, 7
- estimating SOA projects, 8
- organizational maturity, 7
- scope of SOA transformation, 5
- SOA Center of Excellence (CoE), 8

ROI (return on investment), 314

roles in service factories, 97-113

Roosevelt, Theodore, 329

Ross, Jeanne W., 69

S

Saint-Exupery, Antoine de, 114

scope

- of operations, 297-299
- of SOA transformation, 5

SDLC (system development lifecycle).

- See also lifecycle management*
- business requirements and service identification control point, 172
- Ideation Communications case study
 - business requirements control gate, 210-213*
 - service development lifecycle controls, 209*
 - solution architecture control gate, 214-217*
- overview, 168-171
- service build control point, 175
- service certification and deployment control point, 176-177
- service design control point, 174-175
- service specification control point, 174
- service test control point, 175-176
- service vitality control point, 177
- solution architecture control point, 173

security architects, 113

Security Management process, 32

service acceptance checklist work product, 187

service architects, 106-107

Service Architecture process, 31

service assemblers, 109-110

Service Assembly process, 31

service broker usage, 317

service build control point, 175

service build management approach work product, 188

service build plan work product, 188-189

service certification, 56

- Ideation Communications case study, 304-305
 - certification detail, 355-356*
 - certification header, 355*
 - overview, 353-355*
 - service certification detail, 306-307*
 - service certification header, 306*

- overview, 176-177
- preparing for, 283-284
- Service Communication Planning capability, 53**
- service construction estimation
 - metrics work product, 189
- service construction monitoring
 - plan work product, 189
- Service Construction process, 32**
- Service Delivery process, 31**
- service deployment approach work product, 190
- service deployment checklist work product, 191
- Service Deployment process, 31**
- service design
 - overview, 31, 167, 174-175
 - preparing for, 279-280
 - service design checklist work product, 190
 - service design walkthrough notes work product, 190-191
- service designers, 108-109**
- service developers, 109-110**
- Service Development domain**
 - capabilities, risks, and remedial work ducts, 179-183
 - overview, 167, 178-179
 - work products
 - control point minutes, 184*
 - executable modeling approach, 185*
 - functional and nonfunctional checklist, 186*
 - functional and nonfunctional requirements, 185-186*
 - overview, 183-184*
 - regulatory compliance approach, 186*
 - regulatory compliance checklist, 187*
 - service acceptance checklist, 187*
 - service build management approach, 188*
 - service build plan, 188-189*
 - service construction estimation metrics, 189*
 - service construction monitoring plan, 189*
 - service deployment approach, 190*
 - service deployment checklist, 191*
 - service design checklist, 190*
 - service design walkthrough notes, 190-191*
 - service granularity, visibility, and accessibility checklist, 191-192*
 - service level agreement, 192*
 - service reusability guidelines, 192-193*
 - service security approach, 193*
 - service security checklist, 193-194*
 - service sourcing policy, 194*
 - service specification, 194-195*
 - service specification checklist, 195*
 - service test plan, 196*
 - service test results, 196*
 - service version management approach, 196*
 - SOA development lessons learned, 198*
 - SOA development performance report, 197*
 - SOA development tools, 197-198*
 - SOA governance policy exemption, 198-199*
 - SOA reuse and ROI report, 199*
- Service Development Lifecycle capabilities, 54-56**
- Service Development Lifecycle Controls capability, 55**
 - Ideation Communications case study, 340
- Service Discovery process, 32**
- Service Education & Training capability, 53**
- Service Elaboration process, 32**
- Service Execution Monitoring capability, 56**
- service externals, 278**
- service factories**
 - Ideation Communications case study
 - Business Requirements control gate, 341-345*
 - service development lifecycle controls, 340*
 - service transformation planning, 337-340*
 - Solution Architecture control gate, 345-348*
 - reasons for creating, 94-97
 - roles and responsibilities in, 97-113
- Service Funding process, 31**
- Service Governance Vitality capability, 53**
- service granularity, regulating**
 - automation, 294-295
 - discrete, standalone operations, 299
 - overview, 287-288

- read-only services, 300
- reusability of operations, 295-297
- scope of operations, 297-299
- service granularity, visibility, and accessibility checklist work product, 191-192
- tasks versus processes, 288-293
- update services, 300-301
- service granularity, visibility, and accessibility checklist work product, 191-192
- service identification, 55, 271-277, 315
- Service Inception process, 32
- Service Integration Maturity Model (SIMM), 7
- service level agreement work product, 192
- Service Modeling process, 31
- service monitoring, 168
- service operational vitality report work product, 57, 205
- Service Operations capabilities, 56-57
- Service Opportunity Identification process, 32
- Service Ownership & Funding capability, 31, 53
 - Ideation Communications case study, 351-353
- Service Portfolio Management capability, 52
- service portfolio work product, 131-132
- service prioritization checklist work product, 132-133
- Service Processes, Organizations, Roles & Responsibilities capability, 52
- service realization, 56, 281-282
- service registrars, 104-105
- service response time, 317
- service reusability guidelines work product, 192-193
- service reuse, 313
- service security approach work product, 193
- service security checklist work product, 193-194
- service sourcing policy work product, 194
- service specification, 56, 174, 278-279
 - service specification checklist work product, 195
 - service specification work product, 194-195
- Service Support, 32, 57
- service test plan work product, 196
- service test results work product, 196
- service testers, 112
- service testing, 31, 168, 175-176
 - preparing for, 282-283
 - service test plan work product, 196
 - service test results work product, 196
 - service testers, 112
- service transformation planning, 52
 - Ideation case study, 161-163
 - Ideation Communications case study, 337-340
- Service Transition process, 33
- service usage data work product, 205-206
- service version management approach work product, 196
- service versioning, 301
 - communicating with service customers, 303
 - designing future-proof services, 302-303
- service vitality point, 177
- service-oriented architecture. *See* SOA
- Services Operations domain capabilities, risks, and remedial work products, 201-202
 - problem incident log, 203-204
 - quality of service goals, 204
 - quality of service monitoring plan, 204
 - quality of service report, 205
 - service operational vitality report, 205
 - service usage data, 205-206
 - SLA compliance report, 206-207
 - SLA monitoring plan, 207
 - technical support approach and targets, 207
 - work products, 203

- services ownership and funding models
 - work product, 130-131
- services selection and prioritization
 - policies work product, 133-134
- services strategic blueprint work product, 134-135
- SGMM (SOA Governance and Management Model)
 - governance processes, 27, 30-33
 - communications*, 30
 - compliance*, 28
 - exceptions and appeals*, 29
 - vitality*, 28
 - illustration, 26
 - overview, 25-26
 - SOA vision, 26-27
- SIMM (Service Integration Maturity Model), 7
- skills, 35
- skip-level communication, 60
- SLA compliance report work product, 206-207
- SLA monitoring plan work product, 207
- SOA (service-oriented architecture), 1
 - benefits of, 2-4
 - business discovery, 227
 - defining SOA vision and strategy*, 228
 - determining existing governance structure*, 229
 - identifying SOA business principles*, 228
 - definition of, 16
 - goals of, 17
 - governance. *See* SOA governance
 - issues and risks in SOA adoption, 4-8
 - Capability Assessment Tool (CAT)*, 7
 - communication*, 7
 - estimating SOA projects*, 8
 - organizational maturity*, 7
 - scope of SOA transformation*, 5
 - SOA Center of Excellence (CoE)*, 8
 - standardization*, 4
 - principles, refining, 236-237
- SOA architects, 105-106
- SOA career paths work product, 135
- SOA communications plan work product, 135-137
- SOA development lessons learned work product, 198
- SOA development performance report work product, 197
- SOA development tools work product, 197-198
- SOA education plan work product, 137
- SOA governance
 - capabilities
 - capability gaps, closing*, 242-245
 - Plan & Organize*, 50-53
 - Program Management Controls*, 53-54
 - Service Development Lifecycle*, 54-56
 - Service Operations*, 56-57
 - characteristics of, 43-46
 - common pitfalls, 46-50
 - core competencies
 - operational management and service monitoring*, 168
 - requirements collection*, 166-167
 - service design*, 167
 - service development*, 167
 - service testing and deployment*, 168
- environment readiness, 229
 - analyzing assessment results*, 231
 - conducting governance planning assessment*, 230
 - tailoring governance planning assessment*, 230
- governance paradigm, 18-22
- Ideation Communications case study
 - business goals*, 38-40
 - company background*, 36-38
 - overview*, 36
- implementation. *See* implementation
- key governance considerations, 20-21
- key governance decisions, 21
- key questions, 17
- maturity levels, 61-62
- mechanisms
 - infrastructure and tools*, 35-36
 - monitors and metrics*, 34
 - organizational change management*, 35

- principles, policies, standards, and procedures*, 33-34
- skills*, 35
- organizations, assessing
 - governance priorities and near-term goals*, 73-83
 - history*, 57-60
 - Ideation Communications case study*, 84-88
 - organization type*, 69-70
 - suitability considerations*, 70
 - symptoms and causes*, 60-68
- overview, 16-18
- pattern for process improvement, 90-91
 - complexity management in*, 71-72, 91-94
 - roles and responsibilities in*, 97-113
 - service factory creation*, 94-97
- Plan & Organize domain (SOA), 50-53, 114-122
 - architecture standards and design patterns work product*, 122-123
 - business agility monitoring plan work product*, 123
 - business and IT financial priorities work product*, 125
 - business flexibility monitoring plan work product*, 124
 - business "heat map" work product*, 124-125
 - business rules and requirements work product*, 125-126
 - business term glossary work product*, 126
 - business vision work product*, 126
 - dependencies in*, 144-146
 - enterprise data model (EDM) work product*, 127-128
 - information systems strategy blueprint work product*, 128
 - IT capacity management plan work product*, 129
 - messaging model (MM) work product*, 127-128
 - operational model work product*, 129-130
 - service portfolio work product*, 131-132
 - service prioritization checklist work product*, 132-133
 - services ownership and funding models work product*, 130-131
 - services selection and prioritization policies work product*, 133-134
 - services strategic blueprint work product*, 134-135
 - SOA career paths work product*, 135
 - SOA communications plan work product*, 135-137
 - SOA education plan work product*, 137
 - SOA governance plan work product*, 137-139
 - SOA governance vitality monitoring plan work product*, 139-140
 - SOA governance vitality report work product*, 140-141
 - SOA reuse and ROI monitoring plan work product*, 141-142
 - SOA roles and responsibilities work product*, 142-143
 - strategic business plan work product*, 143
 - supply chain and procurement processes for SOA work product*, 143
 - technology blueprint work product*, 144
- planning assessment
 - analyzing results of*, 231
 - conducting*, 230
 - defining plans*, 247-249
 - tailoring*, 230
- Program Management Controls domain, 53-54, 146-150
 - business agility monitoring report work product*, 152
 - business flexibility monitoring report work product*, 153
 - dependencies in*, 160
 - project actual costs work product*, 153
 - project audit findings work product*, 153
 - project benefits and ROI monitoring plan work product*, 154
 - project benefits and ROI report work product*, 154
 - project change request work product*, 155
 - project context and scope work product*, 155
 - project control point minutes work product*, 155
 - project dependencies work product*, 156

- project deployment control point checklist work product, 156*
 - project execution monitoring plan work product, 156*
 - project financial approval control point checklist work product, 157*
 - project lessons learned work product, 157*
 - project outline approval control point checklist work product, 158*
 - project requirements and scope control point checklist work product, 158*
 - project status report work product, 158*
 - project workloads estimate work product, 159*
 - vendor assessment guidelines for SOA work product, 159*
 - vendor guidance notes for SOA work product, 160*
 - SGMM (SOA Governance and Management Model)
 - exceptions and appeals processes, 29*
 - governance processes, 27-30*
 - illustration, 26*
 - overview, 25-26*
 - processes to be governed, 30-33*
 - SOA vision, 26-27*
 - vision, 42-43
 - SOA Governance and Management Model. *See* SGMM
 - SOA governance leads, 100-101
 - SOA governance plan work product, 137-139
 - SOA governance policy exemption work product, 198-199
 - SOA governance team, 222
 - SOA governance vitality monitoring plan work product, 139-140
 - SOA governance vitality report work product, 140-141
 - SOA planning assessment. *See* assessing organizations
 - SOA reuse and ROI monitoring plan work product, 141-142
 - SOA reuse and ROI report work product, 199
 - SOA roles and responsibilities work product, 142-143
 - SOA Strategy process, 31
 - solution architecture
 - Ideation Communications case study, 214-217, 345-348
 - overview, 173
 - preparing for, 277-278
 - staffing, 221-222
 - standalone operations, 299
 - standards, 4, 33-34
 - startup activities, 224
 - conducting kickoff meetings, 227
 - conducting planning workshops, 226
 - gathering current state documentation, 225-226
 - state, gathering current state documentation, 225-226
 - strategic business plan work product, 143
 - strategy, defining, 228
 - supply chain and procurement processes for SOA work product, 143
 - system development lifecycle. *See* SDLC
- ## T
- tasks
 - automation, 294-295
 - evaluating, 268
 - versus processes, 288-293
 - TCO (total cost of ownership), 313
 - technical support approach and targets work product, 207
 - technology blueprint work product, 144
 - Technology Transformation Planning capability, 52
 - testing, 168
 - preparing for, 282-283
 - service test point, 175-176
 - timeline for SOA governance implementation, 223
 - tools, 35-36
 - defining, 245-247
 - implementing, 255-256
 - top-down approach to defining goals, 311-313

top-down requirements analysis,
267-270
total cost of ownership (TCO), 313
transition plans, 250-251, 266
Treacy, Michael, 310
triage process (service identification),
271-277

U-V

update services, 300-301
updating business principles, 236-237

vendor assessment guidelines for
SOA work product, 159
vendor guidance notes for SOA work
product, 160
Vendor Management capability, 54
versioning, 301
communicating with service
customers, 303
designing future-proof services, 302-303
vision, 26-27
defining, 228
setting, 42-43
vitality
feedback process, 321-324
assess governance capability phase, 323-324
develop governance enhancements phase, 324
evaluate management performance phase,
322-323
goals and measurements
application portfolio rationalization, 316
business to IT alignment, 314
customer intimacy, 312
defining with bottom-up approach, 313-317
defining with top-down approach, 311-313
governance process compliance, 315
governance process maturity, 316
maturity index, 315
overview, 310
product leadership, 312
reference architecture, 314
ROI (return on investment), 314
service broker usage, 317
service identification technique, 315

service response time, 317
service reuse, 313
TCO (total cost of ownership), 313
Ideation Communications case study,
325-328, 357-359
overview, 28, 309
preparing for, 285
reporting, 318-321
service vitality point, 177
Vonnegut, Kurt, 178

W-X-Y-Z

Weill, Peter, 69
Whitehead, Alfred North, 201
Wiersema, Fred, 310
Wilde, Oscar, 1, 209
work products
architecture standards and design
patterns, 122-123
business agility monitoring plan, 123
business agility monitoring report, 152
business and IT financial priorities, 125
business flexibility monitoring plan, 124
business flexibility monitoring
report, 153
business "heat map," 124-125
business rules and requirements, 125-126
business term glossary, 126
business vision, 126
control point minutes, 184
defined, 92
dependencies in, 144-146, 160
deprecated or decommissioned
services, 203
enterprise data model (EDM), 127-128
executable modeling approach, 185
functional and nonfunctional
checklist, 186
functional and nonfunctional
requirements, 185-186
information systems strategy
blueprint, 128
IT capacity management plan, 129
messaging model (MM), 127-128

- operational model, 129-130
- overview, 183-184
- problem incident log, 203-204
- project actual costs, 153
- project audit findings, 153
- project benefits and ROI monitoring plan, 154
- project benefits and ROI report, 154
- project change request, 155
- project context and scope, 155
- project control point minutes, 155
- project dependencies, 156
- project deployment control point checklist, 156
- project execution monitoring plan, 156
- project financial approval control point checklist, 157
- project lessons learned, 157
- project outline approval control point checklist, 158
- project requirements and scope control point checklist, 158
- project status report, 158
- project workloads estimate, 159
- quality assurance levels, 71-72
- quality of service goals, 204
- quality of service monitoring plan, 204
- quality of service report, 205
- regulatory compliance approach, 186
- regulatory compliance checklist, 187
- service acceptance checklist, 187
- service build management approach, 188
- service build plan, 188-189
- service construction estimation metrics, 189
- service construction monitoring plan, 189
- service deployment approach, 190
- service deployment checklist, 191
- service design checklist, 190
- service design walkthrough notes, 190-191
- service granularity, visibility, and accessibility checklist, 191-192
- service level agreement, 192
- service operational vitality report, 205
- service portfolio, 131-132
- service prioritization checklist, 132-133
- service reusability guidelines, 192-193
- service security approach, 193
- service security checklist, 193-194
- service sourcing policy, 194
- service specification, 194-195
- service specification checklist, 195
- service test plan, 196
- service test results, 196
- service usage data, 205-206
- service version management approach, 196
- services ownership and funding models, 130-131
- services selection and prioritization policies, 133-134
- services strategic blueprint, 134-135
- SLA compliance report, 206-207
- SLA monitoring plan, 207
- SOA career paths, 135
- SOA communications plan, 135-137
- SOA development lessons learned, 198
- SOA development performance report, 197
- SOA development tools, 197-198
- SOA education plan, 137
- SOA governance plan, 137-139
- SOA governance policy exemption, 198-199
- SOA governance vitality monitoring plan *in SOA Plan & Organize domain*, 139-140
- SOA governance vitality report, 140-141
- SOA reuse and ROI monitoring plan, 141-142
- SOA reuse and ROI report, 199
- SOA roles and responsibilities, 142-143
- technical support approach and targets, 207
- vendor assessment guidelines for SOA, 159
- vendor guidance notes for SOA, 160
- work-product-based approach to complexity management**, 71-72, 92-94
- workshops, planning workshops**, 226