Oracle® Practitioner Guide

Determining ROI of SOA through Reuse

Release 3.2

E14487-03

February 2012

ORACLE®

Determining ROI of SOA through Reuse, Release 3.2

E14487-03

Primary Author: Stephen G. Bennett

Contributing Author: Sharon Fay

Contributor: Bob Hensle, Dave Chappelle, Cliff Booth, Anbu Krishnaswamy Anbarasu, Mark Wilkins

Contents

List of Tables

List of Figures

Send Us Your Comments

Determining ROI of SOA through Reuse, Release 3.2

E14487-03

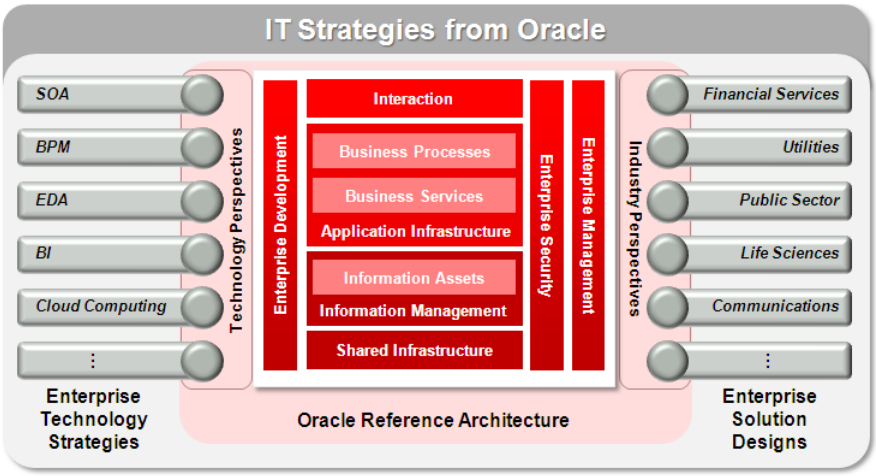
Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this document?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us at its_feedback_ww@oracle.com.

Preface

IT Strategies from Oracle (ITSO) is a series of documentation and supporting collateral designed to enable organizations to develop an architecture-centric approach to enterprise-class IT initiatives. ITSO presents successful technology strategies and solution designs by defining universally adopted architecture concepts, principles, guidelines, standards, and patterns.



ITSO is made up of three primary elements:

- **Oracle Reference Architecture (ORA)** defines a detailed and consistent architecture for developing and integrating solutions based on Oracle technologies. The reference architecture offers architecture principles and guidance based on recommendations from technical experts across Oracle. It covers a broad spectrum of concerns pertaining to technology architecture, including middleware, database, hardware, processes, and services.
- **Enterprise Technology Strategies (ETS)** offer valuable guidance on the adoption of horizontal technologies for the enterprise. They explain how to successfully execute on a strategy by addressing concerns pertaining to architecture, technology, engineering, strategy, and governance. An organization can use this material to measure their maturity, develop their strategy, and achieve greater levels of success and adoption. In addition, each ETS extends the Oracle Reference Architecture by adding the unique capabilities and components provided by that particular technology. It offers a horizontal technology-based perspective of ORA.
- **Enterprise Solution Designs (ESD)** are industry specific solution perspectives based on ORA. They define the high level business processes and functions, and the software capabilities in an underlying technology infrastructure that are

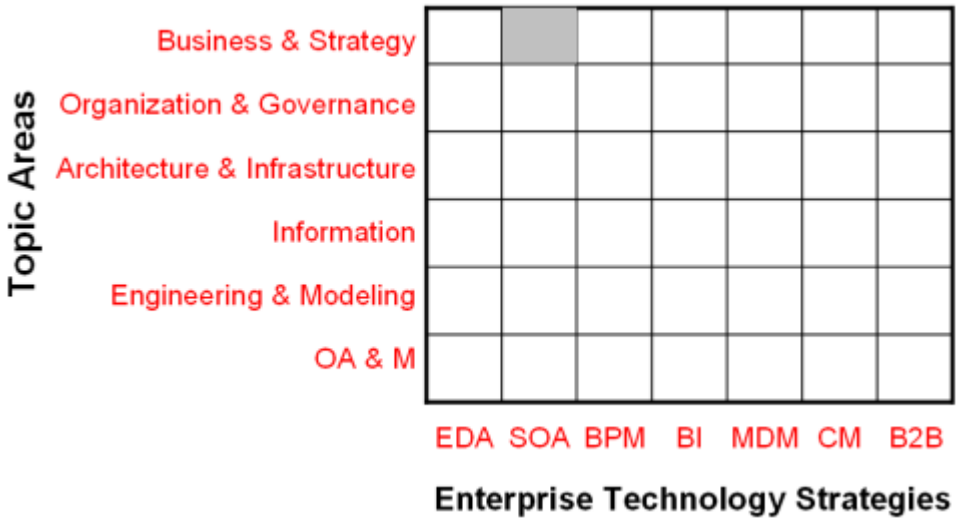
required to build enterprise-wide industry solutions. ESDs also map the relevant application and technology products against solutions to illustrate how capabilities in Oracle's complete integrated stack can best meet the business, technical, and quality of service requirements within a particular industry.

This document is part of a series of documents that comprise the SOA Enterprise Technology Strategy, which is included in the IT Strategies from Oracle collection.

Please consult the ITSO web site for a complete listing of SOA and ORA documents as well as other materials in the ITSO series.

Document Purpose

A Practitioner's Guide provides insight and guidance when working with a particular type of technology and address the common concerns faced by enterprises and practitioners.



This Practitioner's Guide provides an approach for estimating the value of the various software assets contained in a typical SOA portfolio. This approach can assist in building the business case for, and determining the impact of, Service-Oriented Architecture. It should be noted that this guide describes an SOA asset valuation process that focuses primarily on development savings and does not cover the operational aspects as well as the business savings and opportunities garnered from SOA

Audience

The primary audience for this guide are IT managers, enterprise architects, and other stakeholders who are responsible for building the business case and determining the impact of Service-Oriented Architecture.

Document Structure

This document is organized into the following sections.

Chapter 1 - provides an overview of an approach to determine the ROI of SOA through resue.

Chapter 2 - provides an illustration in how to calculate an estimated reuse value of an asset.

Chapter 3 - provides an approach in prioritizing investments in reusable services and other SOA assets.

Chapter 4 - provides a summary of the document.

Appendix A - provides a snapshot of a sample worksheet.

Appendix B - provides a quick reference on where to find further information.

How to Use This Document

This document should be read by everyone that is interested in learning an approach for estimating the value of the various software assets contained in a typical SOA portfolio

Refer to the ***ORA Glossary*** document for descriptions of key terms.

Conventions

The following typeface conventions are used in this document:

Convention	Meaning
boldface text	Boldface type in text indicates a term defined in the text, the glossary, or in both locations.
italic text	Italics type in text indicates the name of a document or external reference.
underline text	Underline text indicates a hypertext link.

In addition, the following conventions are used throughout the IT Strategies from Oracle documentation:

"Service" v. "service" - In order to distinguish the "Service" of Service Oriented Architecture, referred to throughout the document series, the word appears with its initial letter capitalized ("Service"), while all other uses of the word appear in all lower-case (e.g. "telephone service"); exceptions to this rule arise only when the word "service" is part of a name, such as, "Java Message Service" ("JMS"), "Web Service", etc.

1

Overview

IT organizations have at their disposal software assets that, if systematically reused in the transformation to a **Service-Oriented Architecture (SOA**), can potentially save hundreds of thousands or even millions of dollars each year by increasing development productivity and decreasing software maintenance costs. The resulting performance improvements can have a significant impact on business agility and the ability to respond quickly to competitive challenges.

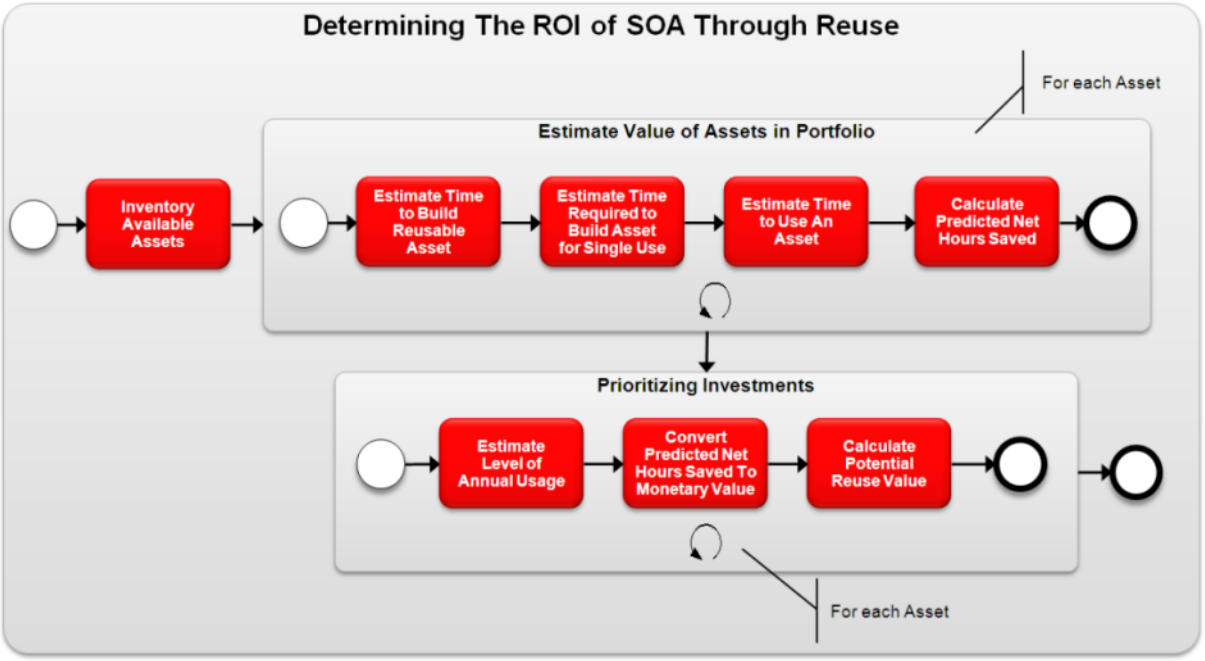
The transformation to SOA involves building and deconstructing applications into a matrix of approved, discreet, loosely coupled, and highly reusable Services. The governance measures in place to manage that transformation and ensure the maximum reuse of Services and related artifacts must include the means to assess, track, and communicate the value of that reuse. This provides the feedback critical to determining whether the SOA is on track to meet organizational goals. That information then guides investment and other decisions about the evolution of the SOA.

An organization's SOA portfolio of software assets includes Services and other supporting artifacts that comprise the SOA. In order to gauge the return on investment for the reuse of Services and other assets, an organization must assess the potential value of key assets within its portfolio, continuously measure the impact of reuse on development productivity, and determine the ultimate impact on its bottom line.

This guide focuses on the stages of the Service lifecycle during which Services are designed and created. It provides an approach for estimating the value of the various software assets contained in a typical portfolio. This approach can be used by IT managers, enterprise architects, and other stakeholders in building the business case for, and determining the impact of, Service-Oriented Architecture. It should be noted that this guide describes an asset valuation process that focuses primarily on development savings. A comprehensive asset valuation process should also consider the operational aspects as well as the business savings and opportunities garnered from SOA.

The asset valuation process should expand incrementally, beginning with the simpler tasks: those assets that are of the most obvious relevance and value to the SOA. For example, common security components or customer information services may already be in use across multiple applications and projects. The asset valuation process should start with these Services and then expand to encompass other assets, providing an increasingly comprehensive and accurate picture of the asset portfolio's overall value.

Figure 1–1 Determining The ROI of SOA Through Reuse



The process and calculations described in this document distill years of experience with customers as well as recognized industry best practices. They were designed to estimate the potential reuse value of each asset and to project the expected return on investment (ROI) on an asset portfolio. On an individual basis, some organizations using the asset valuation process described in this document have recorded millions of dollars in reuse savings.

The companion Excel® workbook assists in establishing the value of an asset portfolio, and the following steps walk through the worksheet entries and explain the logic behind the calculations. At each step, the variables used in the calculations can be modified to reflect specific situations at any organization.

2

Estimating Value of Assets in the Portfolio

A reusable asset can be defined as any artifact of the software development or SOA lifecycle, including but not limited to: Services (Web services, data services, etc.), business processes, architecture models, applications, frameworks, components, WSDL files, XML schemas, DTDs, quite literally any code, metadata, or supporting artifact that is relevant to the SOA. If it has value, that value can be extended through reuse and that reuse value should be tracked and measured.

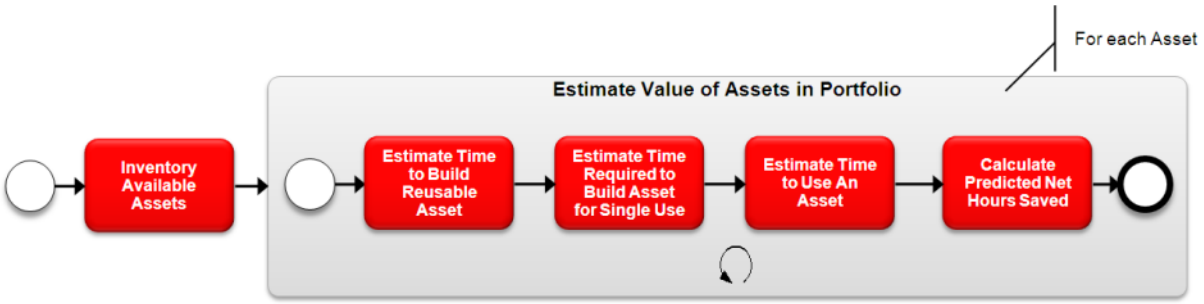
Within the confines of this focused approach, an asset's estimated value is based on the development costs avoided by reusing rather than recreating it. That measurement is referred to here as **Predicted Net Hours Saved**. By reusing existing assets, an organization avoids both the costs of repeatedly developing the same functionality and the costs of maintaining different implementations of the same functionality. Reuse consolidates functionality and reduces redundancy. The more that reuse occurs, the greater the savings.

The formula used to calculate avoided costs is simple and straightforward. The first part of the equation estimates the amount of time a developer would spend creating the asset for a single use. Next, the formula deducts the time spent to reuse an existing asset on a project. The result is an estimate of the net hours saved by reusing it.

Figure 2–1 Asset Estimated Value Formula

The formula then considers the level of annual usage for each asset and converts the total predicted net hours saved to a monetary value. The result is an estimate of the potential value of existing assets in the organization's portfolio. The valuation process can be seen below.

Figure 2–2 Estimate Value of Assets in Portfolio



2.1 Inventory Available Assets

The first step in the valuation process is to inventory and organize available assets that will support the tranistion to SOA. Given the sheer volume of assets that may be available for reuse, the inventory process should proceed incrementally, identifying and categorizing key assets. Those that are in the greatest demand and are the most frequently used. The process can then expand to encompass other assets, as resources allow.

The assets and supporting artifacts addressed in this document fall into three basic categories.

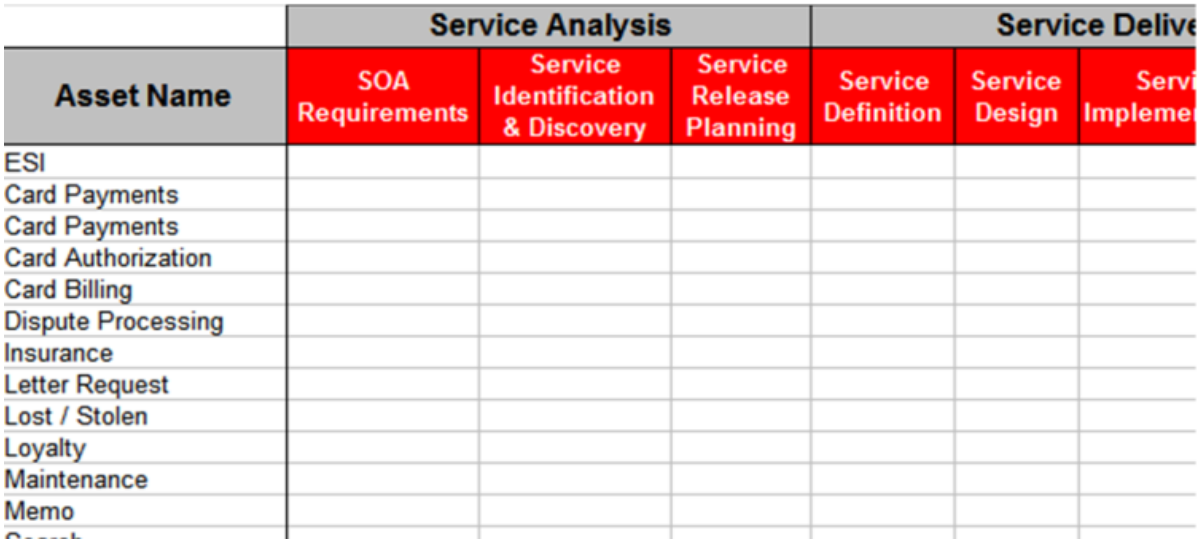
- **Services** - Services and other assets exposed through contracts/interfaces typically have higher production and overhead costs. For example, it takes time to gather and generalize requirements, parameterize the interface, and test under a broad spectrum of use cases. However, Services and similar assets also offer the most significant overall reuse savings. Developers can easily access and understand a Service's contract and interface and test its functionality, avoiding much of the design, development, and testing necessary for other types of assets. In addition, because there is a single implementation of the Service's functionality, ongoing maintenance costs are greatly reduced. (See *ORA SOA Foundation* document for detailed description of a Service)
- **Black-box code assets** - The term black-box indicates that an asset's code implementation is hidden from the end-user. Unlike Services, which typically have one deployed instance that is called by multiple users, black-box assets are incorporated into the developer's code base, resulting in multiple deployed instances. Rate calculators and security frameworks are typical examples of black-box assets, which are distributed as binaries or as obfuscated objects. Clearly defined and documented interfaces serve as the communication channel between the asset and its surrounding environment. While the production and overhead costs for developing black-box assets are similar to those for Services, any reuse savings are reduced by testing and maintenance costs incurred when the black-box asset is incorporated into the consuming developer's code base.
- **White-box code assets** - The term white-box is used to indicate that the code implementation of an asset is visible to end-users and that the source code can be modified when it is incorporated into the developer's code base. Open-source assets, such as Struts, are white-box by definition. White-box assets offer the lowest level of return in a reuse program. While they can improve development productivity, their value in terms of maintenance savings is limited because their use can result in the deployment of multiple variations of an asset, each of which must be individually supported. If a mandatory change occurs to the core

functionality of a white-box asset, each deployed variation will have to be examined, modified, and retested.

While black-box and white-box assets have traditional been aligned with component based reuse, there is still value is considering the reuse of these asset types in an SOA environment due to their predicted use by Services themselves.

The companion Excel® workbook can be used to capture the different asset types in the appropriate worksheets (Services, Black Box, and White Box)

Figure 2–3 Inventory Available Assets



	Service Analysis			Service Delive		
Asset Name	SOA Requirements	Service Identification & Discovery	Service Release Planning	Service Definition	Service Design	Servi Impleme
ESI						
Card Payments						
Card Payments						
Card Authorization						
Card Billing						
Dispute Processing						
Insurance						
Letter Request						
Lost / Stolen						
Loyalty						
Maintenance						
Memo						

2.2 Estimate Time To Build Reusable Asset

Within each worksheet (service, black box, white-box) enter the time invested in producing each reusable asset and any supporting artifacts. This number should reflect the total development effort for each asset. For convenience, the time estimates are broken down into individual lifecycle stages.

The table below provides an example of the time invested in producing a reusable security service that authenticates and validates user IDs.

Table 2–1 Time Invested in Producing a Reusable Security Service (Example)

Service Analysis	
SOA Requirements	120 Hours
Service Identification & Analysis	8 Hours
Service Release Planning	40 Hours
Service Delivery	
Service Definition	24 Hours
Service Design	80 Hours
Service Implementation	80 Hours
Service Test	120 Hours
Total	472 Hours

Following the example in the table, conduct the same exercise for assets in your portfolio. The cumulative total of this measurement will be automatically calculated. Note that the numbers entered will be estimates. These estimates should be correct and should be conservative. For example, if the estimate ranges between three and six weeks, use the three-week figure. If an estimated value for each phase of the lifecycle is unavailable, enter an estimate of the total time.

Bear in mind that reuse value can accumulate quickly, especially in a well-managed reuse environment. So in keeping with the idea that the asset valuation process should start small, with the low-hanging fruit, even a conservative, "good enough" estimate of the time invested in producing a reusable asset can result in compelling evidence of the value of reuse.

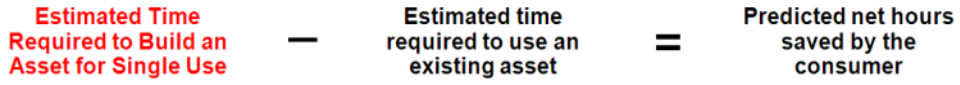
2.3 Estimated Time Required To Build Asset For Single Use

Table 2–2 Production Investment for Reusable Assets

Asset	Production Investment
Services	200%
Black-box Assets	200%
White-Box Code Assets	100% - 200%

Building and/or harvesting a Service or other reusable asset typically takes more time and effort than building an asset for single use. This extra time and effort is called the Production Investment and it includes the time and effort involved in the development, packaging, and documentation necessary to enhance an asset's reusability and availability across the widest possible audience. For example, a developer might invest 540 hours creating a function for use on a project. An organization will spend approximately twice the amount of time-200%-building a reusable Service

The reuse savings estimate for each asset is based on the time that a developer would have spent developing equivalent functionality for a single use. The worksheet that accompanies this document uses the total development hours and the Production Investment values to estimate that time and effort..



For example, Table 2–1 indicates a total investment of 472 hours to produce the reusable Service. If the average production investment for a reusable Service is 200 percent (that is, twice the time it takes to build the equivalent functionality in a single-use asset), a developer would have spent 236 hours building a single-use version of the Service. That figure (236 hours) provides the basis for calculating the estimated reuse savings for the Service.

Table 2–2 lists the Production Investment[1] for various categories of reusable assets. Notice that in most cases the Production Investment exceeds 100 percent, reflecting the additional effort required beyond what is needed to build a single-use asset.

Given that the reuse of a white-box asset assumes some level of modification, the ease with which the consuming developer can make the necessary modifications to the asset is a significant factor in that asset's reusability. Additional documentation and

other resources provided to the developer to facilitate white-box asset modification are part of the Production Investment. For white-box assets, this investment ranges from 100 percent (no additional effort is invested in improving the asset's reusability) to 200 percent. Obviously, there is a direct correlation between the effort invested in making an asset reusable and the effort required to reuse it. That reuse effort is discussed in the next section.

The values above are provided as a baseline. Few organizations actually measure the Production Investment associated with the reusable assets in their portfolios until they have established a managed reuse program. For the purposes of the accompanying workbook, these values may be used as is, or may be adjusted to reflect organizational experience.

The next step is to calculate and subtract the Consumption Factor, the time that a developer will spend reusing the asset.

2.4 Estimate Time To Use An Asset

Table 2–3 Consumption Factor for Different Types of Assets

Asset	Production Investment
Services	5%
Black-box Assets	20%
Domain-specific black-box code assets	5%
White-box code Assets	40% (< 25% code modifications)
White-box code Assets	90% (> 25% code modifications)

Reuse is not free. It takes time to evaluate and then reuse an asset in a project. This time is estimated using the Consumption Factor.

Table 2–3 displays the Consumption Factors associated with various asset categories. As previously discussed, a developer would have spent 236 hours building a single-use version of the Service in our example. The consumption factor for Services is 5 percent. (In this context, consumption refers to the design-time incorporation of a Service call in the development of an application.) Applying this consumption factor

[1] Compiled from data presented in Managing Software Reuse, by Wayne C. Lim (Prentice Hall, 1998, p. 122). Additional resources include:

Bardo. T., et al. March 1996. CORE: A product line success story. The Journal of Defense Software Engineering. Vol. 9, No. 3. pp. 24¬28.

Margano, J. and Lindsey, L. May 1991. Software Reuse in the Air Traffic Control Advanced Automation System. Joint Symposia and

Workshops: Improving Software Process and Competitive Position.

Margano, J. and Rhoads, T. May 1992. Software Reuse Economics: Cost Benefit Analysis on a Large-Scale ADA Project. Proceedings of the International Conference on Software Engineering. pp. 338¬348.

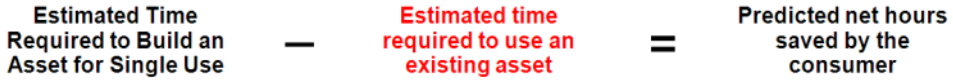
Ramesh, M. and Rao, H. August 1994. Software Reuse: Issues and Example. Decision Support Systems. pp. 57¬77.

Tracz, W. January 1988. Software Reuse Myths. ACM SIGSOFT Software Engineering Notes. pp. 17¬21.

Poulin, J. 1997. Measuring Software Reuse: Principles, practices, and economic models. Addison-Wesley.

Selby, R. 1989. Quantitative Studies of Software Reuse. Software Reusability Vol. 2. Biggerstaff, T. and Perlis, A., eds. Addison¬Wesley.

to the previously determined 236 hours figure produces an estimate of 12 hours to reuse the service.



Oracle's experience with utilizing a registry/repository to discover assets and their relationships confirms the findings of a number of studies with regard to the Production Investment and Consumption Factors associated with reuse. In Table 2–3 note that there are two entries for white-box assets. The first reflects the Consumption Factor that applies if less than 25 percent of an asset's code is modified when the asset is reused. The second entry reflects the Consumption Factor that applies when more than 25 percent of the asset's code is modified. As a rule of thumb, the Production Investment and Consumption Factor for white-box assets are inversely proportional. That is, if no effort is made during production to make an asset reusable (Production Investment ~100 percent), the developer will spend more time evaluating and modifying the asset in order to reuse it, resulting in a higher Consumption Factor (Consumption Factor ~90 percent).

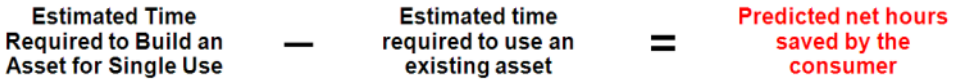
Table 2–3 also includes two entries for black-box code assets and their associated artifacts. The first entry reflects the general Consumption Factor for a black-box code asset, while the second reflects the Consumption Factor that applies if the asset is produced and consumed within the same domain. The advantage developers have when using domain-specific reusable assets is reflected in these Consumption Factors. It takes less time and effort to understand and reuse domain-specific assets if the developer is familiar with the domain.

As with the Production Investment values described, Consumption Factor values are provided only as a baseline and may be adjusted to reflect organizational experience.

2.5 Predicted Net Hours Saved

The difference between the time a developer would spend building an asset for single use and the time needed to use a reusable version of that asset is referred to as Predicted Net Hours Saved. This represents the development cost avoided through reuse of the asset.

The Service in our example would have taken about 236 hours to develop for single use. Based on the calculations described above, a developer will take an estimated 27 hours to reuse the security Service. The difference is the Predicted Net Hours Saved value for the Service.



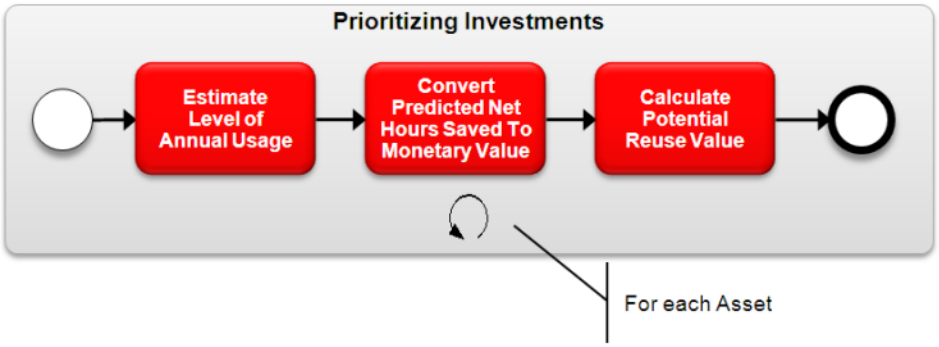
3

Prioritizing Investments

The previous section illustrates how to calculate an asset's estimated reuse value. Those responsible for building a business case for reuse, or for prioritizing investments in reusable Services or assets, should use the following steps below to translate the reuse value into the expected ROI.

Establishing the expected ROI of assets within the portfolio requires consideration of annual usage of each asset, and the conversion of the total Predicted Net Hours Saved to a monetary value. (See Figure 3–1)

Figure 3–1 Prioritizing Investments



3.1 Estimate Level of Annual Usage

In any given year, most organizations have a number of development projects in their pipeline. It is likely, however, that only a subset of these projects will actually have an opportunity to reuse assets in the organization's portfolio. For example, of the 100 projects an organization may have funded for next year, 25 percent may deal with facilities- building and property updates. Reuse of software assets in projects of this nature may be impractical or inapplicable. In this example, the potential for asset reuse is possible and appropriate in only 75 projects (100 minus 25 percent).

The next step is to estimate how many times each asset will be reused. It is unlikely that every project with access to the asset portfolio will leverage every available asset. For example, perhaps only one-third of the projects that represent opportunities for reuse will be able to reuse the example service. This translates to 25 annual reuse opportunities (one-third of 75).

3.2 Convert Predicted Net Hours Saved To Monetary Value

The term Burden Rate refers to the hourly overhead cost associated with each of the various asset-consumer roles. The average Burden Rate for software developers may be different than the average Burden Rate for business analysts. Burden Rates should be standard for each organization. The worksheet uses the Burden Rate to calculate the Potential Reuse Value for each asset.

3.3 Calculate Potential Reuse Value

The Projected Annual Saving Summary worksheet totals the savings from all of the worksheets. This total should be used as the basis for justifying your reuse program and as a target goal for that program.

3.4 Track Progress & Savings

An enterprise registry/repository with the appropriate capabilities is essential in tracking progress toward organizational SOA and reuse goals. A registry/repository should track each asset's value, state (defined, designed, implemented, …) and then, as assets are used, the registry/repository should gather the necessary data to generate reports that detail assets' values as used in the SOA initiative.

This provides a way for organizations to compare estimated ROI to actual returns and to actual returns.

4

Summary

While reusable assets may vary in their size, scope, and purpose, their estimated valuation-and the accurate reporting of their actual value-are essential in guiding SOA governance efforts to ensure sustainable alignment with business goals.

Just as asset valuation information is vital in guiding SOA governance efforts, those efforts are essential for increasing the value of the assets by insuring their alignment with, and support of, policies and standards established at the architectural, IT, and corporate levels. Ultimately, an asset's value is determined not solely by its reusability or reuse, but also by the contribution it makes in moving the organization toward its business goals. To that end, the importance of effective SOA lifecycle governance in guiding the design, development, and use of assets cannot be overstated.

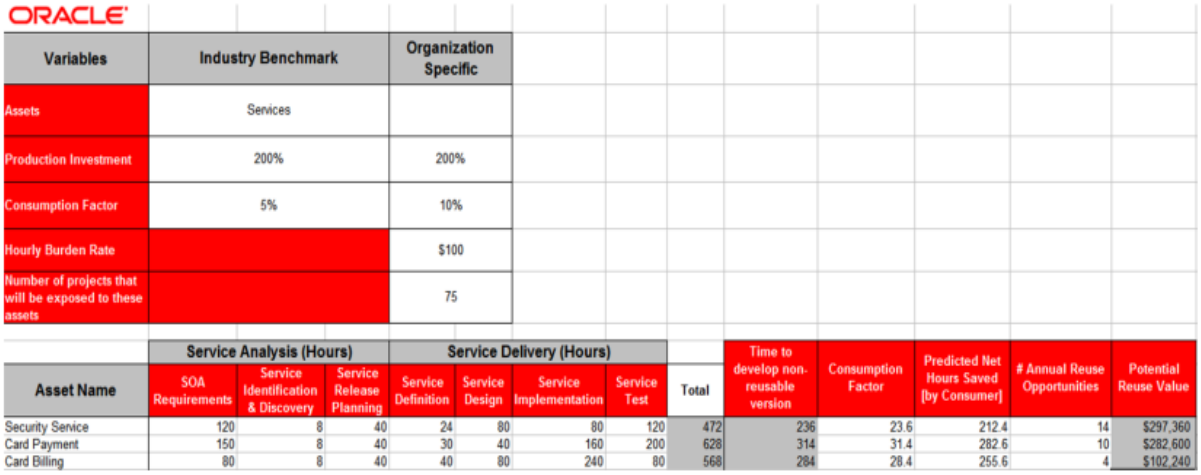
While the enterprise portfolio of Services and other software assets plays a vital role in an SOA, it represents one aspect of the overall value of a Service-Oriented Architecture. Given the SOA environment's unique nature, a variety of factors contribute to the ultimate ROI of SOA, including the sharing of Services in the operational environment and the savings and benefits garnered from a business perspective.

The approach outlined in this guide, when used in conjunction with a robust enterprise registry/repository, will provide an organization with the information necessary to make informed decisions regarding the management and consolidation of the Services and other software assets in the enterprise portfolio, the assignment of those assets for use/reuse in projects, and the ongoing, proactive governance of the SOA.

A

Sample Worksheet

Figure A–1 Sample Worksheet



ORACLE

Variables	Industry Benchmark	Organization Specific
Assets	Services	
Production Investment	200%	200%
Consumption Factor	5%	10%
Hourly Burden Rate		$100
Number of projects that will be exposed to these assets		75

	Service Analysis (Hours)			Service Delivery (Hours)									
Asset Name	SOA Requirements	Service Identification & Discovery	Service Release Planning	Service Definition	Service Design	Service Implementation	Service Test	Total	Time to develop non-reusable version	Consumption Factor	Predicted Net Hours Saved (by Consumer)	# Annual Reuse Opportunities	Potential Reuse Value
Security Service	120	8	40	24	80	80	120	472	236	23.6	212.4	14	$297,360
Card Payment	150	8	40	30	40	160	200	628	314	31.4	282.6	10	$282,600
Card Billing	80	8	40	40	80	240	80	568	284	28.4	255.6	4	$102,240

B

Further Reading

The ***IT Strategies From Oracle*** series contains a number of documents that offer insight and guidance on many aspects of technology. In particular, the following documents pertaining to Determining ROI of SOA Through Reuse may be of interest:

Creating an SOA Roadmap - The purpose of this document is to describe a repeatable process for constructing an SOA Roadmap. The process described follows the standard four steps used within the industry to create roadmaps i.e. establish the current state, define the future vision, analyze the gap, and define the phases and schedule of the roadmap.

A Framework for SOA Governance - Provides an approach which eases the transition and on-going execution of an enterprise's service-orinted architecture (SOA) by providing a means to reduce risk, maintain business alignment, and show the business value of SOA investments.

Refer to the ***ORA Glossary*** document for descriptions of key terms.