



# Delivering the Business Benefits of Service-oriented Architecture (SOA)

Dispelling the Myths, Clarifying the Benefits and Creating the Roadmap  
Includes Real Life Case Studies

A CSC White Paper







# Delivering the Business Benefits of SOA:

Dispelling the Myths, Clarifying the Benefits and Creating the Roadmap

**AUTHOR:** David Campbell BSc(IT), Enterprise Architect, CSC Australia

## Contents

Introduction .....	4
What is SOA?.....	4
Why is SOA so Important? .....	5
What makes a good Business Service? .....	5
Enabling Technologies .....	6
What are the Business Benefits of SOA? .....	7
Which Organisations will Benefit most from SOA? ...	8
Recognising the Stumbling Blocks .....	9
Avoiding the Pitfalls .....	9
Conclusion .....	11
References.....	11

## Introduction

While the idea behind Service-oriented Architecture (SOA) is simple; its implementation is anything but - particularly in a legacy environment.

This White Paper looks at the issues preventing organisations from gaining the true business benefits of SOA. It provides a definition of SOA, discusses its substantial potential benefits and highlights the stumbling blocks organisations typically hit when trying to implement it. It also describes methodologies for avoiding these pitfalls and implementing SOA successfully.

The author has drawn from CSC's extensive global experience helping large organisations maintain a competitive edge and reduce the cost of doing business.

## What is SOA?

SOA is an enabling architecture based on reusable building blocks called business services. Unlike previous architectures, SOA focuses on business processes, rather than technical components.

This means that, using SOA and its **enabling technologies**<sup>1</sup>, software developers can deliver business functionality as a set of

1 - See Page 6

services that can be deployed and combined to address new business needs at minimum cost and with minimum delay.

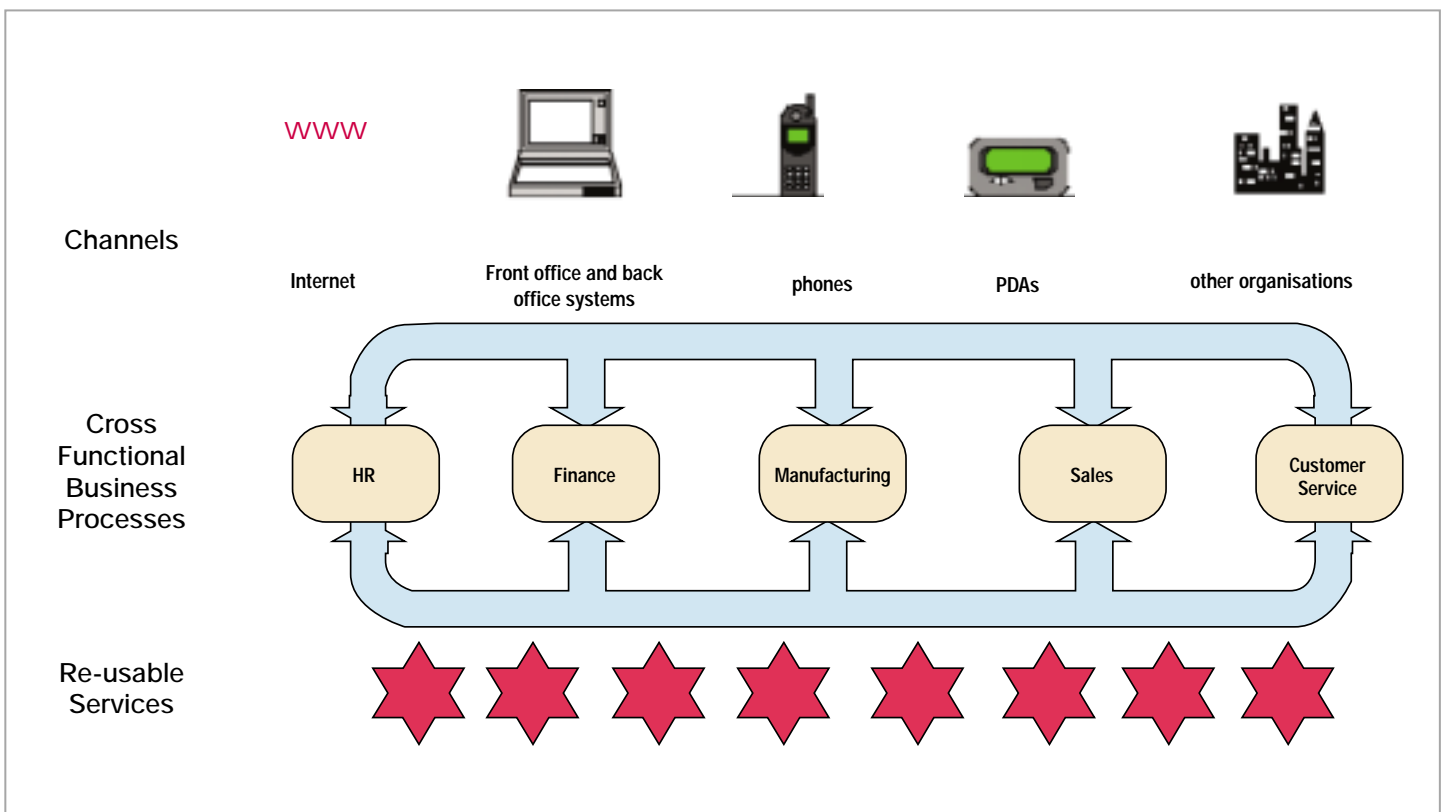
### How does it do this?

Conceptually, SOA consists of Business Applications, Business Services, IT Services and Data Services. Using an SOA approach, most of the common business logic resides in Business Services while the business logic specific to the business process at hand is in the Business Application. Ideally, most of the Business Application's work is in co-ordinating Business Services to achieve the overall business process.

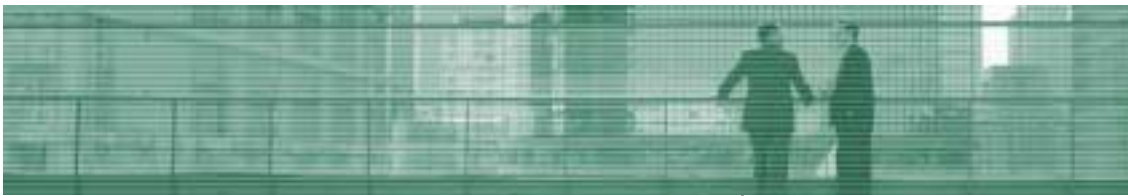
SOA allows developers to construct these Service-oriented Business Applications (SOBAs) by co-ordinating Business Services and IT and Data Services to create true end-to-end business processes. SOBAs comprise co-operating services that may span business boundaries.

In other words, SOA enables software applications to be built as collections of collaborating services that interact without regard to each other's platform, data structures, or internal algorithms.

This has recently become possible because the technology required to support such interoperability has 'come of age' and is no longer considered 'bleeding edge'. The enabling technology, which includes Web services, manages all routing issues, differences in technology platforms and differences in message formats. The result is that business systems and underlying services need only minimal knowledge of each other.



Conceptual Service-oriented Architecture



While technology is important as an enabler of SOA, it is not SOA. You can not go out and buy an SOA from a technology vendor. Just as you need a blueprint to build a complex piece of machinery, you need a blueprint to ensure your architecture is created to support your business goals. SOA provides that blueprint.

## Why is SOA so important?

For 20 years systems developers have searched for the Holy Grail: a means of developing systems from reusable building blocks.

In engineering disciplines, building blocks, once developed, can be re-used across systems. This reduces both the cost and the risk of subsequent developments and makes such projects far more predictable to plan, schedule and cost.

However, despite maturing from structured programming to object-oriented system development to component-based system development, no approach has been able to transform system development into a precise discipline that can be planned and controlled.

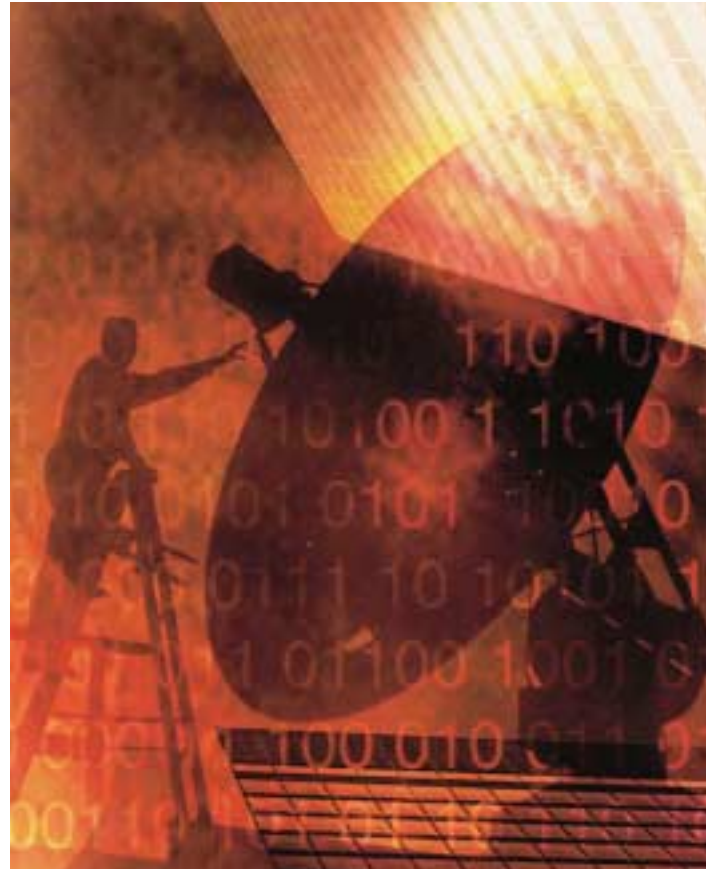
This creates two major business issues. First, because without re-use each new system is practically developed from scratch, IT has been notoriously unable to predict the time and cost of new developments. Second, it has been difficult to adapt existing systems to market changes with anything like the speed most businesses require.

SOA takes a big step towards solving these issues. When you develop services as building blocks, you end up with an assembly kit for creating new systems. This means that system assembly times and therefore costs can be predicted with certainty based on previous system construction efforts. Moreover, services help to reduce the impact of change by hiding their data, technology and

### What makes a good Business Service?

Critical to designing a good set of services is deciding how much work each should do. If the services are too large, your options for assembling them into systems are reduced and you start to lose the benefits of re-use. On the other hand, if the services are too small and therefore numerous, they are hard to integrate and become increasingly difficult to assemble into systems.

A good business service hides its implementation from other business services and business applications. If well designed, a business service can be implemented using COTS products, J2EE or .NET or even using a mixture of all of them as cost and agility dictate. In fact, its implementation can be changed without affecting the processes in which it participates.



algorithms from other services and applications that use them, which in turn results in a faster turn around time.

By using business services as its building blocks SOA overcomes the issues of the component-based development that was fashionable in the 1990s. The problem with components was that they were too fine grained, with each component only handling a very small part of the overall problem, and often focussed more on the technical rather than business parts of the problem.

Business services are much coarser grained. A business service seeks to solve a larger piece of the business problem while minimising the dialogue between the service and its consumers. A business service typically encompasses a complete business sub-process, such as a payment or order process, which can be re-used by other business processes. As an additional benefit, this means it can be specified accurately by non-technical business people, because they only need to understand its business function - not its technical make-up.

SOA technologies can also help manage the problem of how legacy applications, built on disparate technology platforms, talk to each other. They can provide a means by which legacy systems participate in end-to-end business processes without major internal rework, prolonging the life of existing assets.

## Enabling Technologies

Technologies that enable SOA include:

### Web Services

A common misconception is that business services are the same as web services. In fact, while web services are highly interoperable, making them excellent for composite business applications that span business boundaries, they are only one of several options available for SOA. Conversely, adopting web services standards does not automatically give you SOA.

Web Services is based on a messaging protocol and provides standards for interoperability between different software applications running on different platforms. It also provides a language (WSDL) for describing loosely coupled services irrespective of the contexts in which such services will be used and a language for publishing and discovering services (UDDI). Web services make the distinction between a service and the agent that implements it, with the view that new agents can be substituted while continuing to deliver the same service.

### Business Process Management

In implementing business processes, business applications execute rules to determine the flow of control through the application. Traditionally such rules were embedded in application code and could only be maintained by IT staff. In recent years, Business Process Management technologies have emerged that allow people with minimal IT skills to manage business process rules.

Business Process Management products provide a graphical interface for defining business processes by specifying what services to call when and under what conditions to call them. It is possible to achieve a similar result by encoding the processing rules in a business rules engine though in this case the logic flow is still embedded in the business application.

A number of standards have emerged in the business process management area: including Business Process Execution Language for Web Services (BPEL4WS or BPEL). It is an XML based language developed by WC3 that describes how services (specifically web services) are co-ordinated to create processes. In addition, there is a web services standard for process choreography (WSC1).

In competition with the web services standards is ebXML, a standard developed by OASIS as the successor to Electronic Data Interchange (EDI) that specifies a framework for exchange of information between companies doing business with each other (B2B). It also includes its own standard for business process specification.

Many of these standards are still in a state of flux and early adopters can expect standards to continue to evolve over the next few years.

Another approach to integration currently gaining favour is the use of portal technology. Portals provide a level of business process integration by presenting common user access to business applications and services. However, without integration at the back end, the user must understand how the applications and services presented are related and how they are to be used in the course of executing the business process.

### Integration technologies

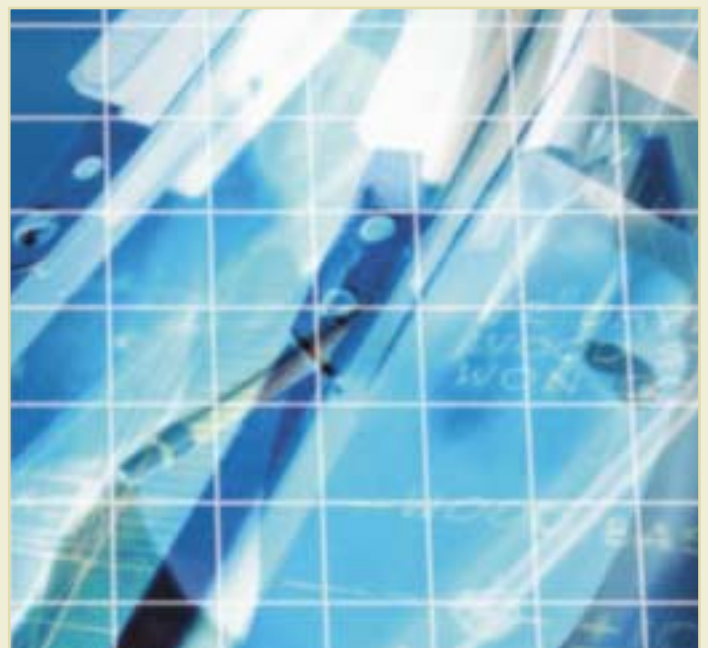
Integration technologies provide the plumbing that enables applications and services to talk to each other. They include:

**Message orientated middleware** - providing support for sending messages between different technology platforms, but without any intelligent routing, data transformation or adaptor connectivity.

**Enterprise Application Integration (EAI) products** - providing the infrastructure link between applications and services running on a variety of platforms. EAI products typically provide intelligent routing based message content, adaptors to a number of technologies and facilities to perform data transformation between applications and services.

**Enterprise Service Bus (ESB) products** - performing a function similar to EAI products but with a smaller footprint and fewer bells and whistles. ESBs claim to have better support for open standards. Their lightweight bus architecture affords them flexibility and scalability and they usually come at a lower cost than full feature EAI products.

**Application suites** - adding another layer on top of EAI and ESB by providing portal functionality, workflow engines, business process co-ordination engines and more sophisticated business to business integration aimed at specific industry segments.





## What are the business benefits of SOA?

### Agility

The ability to quickly and easily couple, uncouple, recombine and restructure your IT systems, in the face of changing market demands, is the most profound business benefit of an SOA.

Because an SOA better aligns technology with business, it enables you to adapt quickly to changing environments. One of the big differences an SOA brings is greater collaboration between business units and system builders when specifying business services, enabling the IT department to keep pace with business imperatives.

This is possible because SOA allows processes to be abstracted from underlying applications and IT systems. Decoupling processes from IT creates enormous business flexibility, allowing business leaders to take greater control of how business is transacted within the enterprise and with partners, suppliers and customers.

This means that, with an SOA, you can provide services to employees, customers and business partners without the time and expense involved in past proprietary efforts. Because everyone follows the same set of standards, you can be responsive, flexible and competitive: you can roll out products and services faster than your competitors and quickly adapt applications to user needs.

### CASE STUDY

CSC is working with a large pharmaceutical company to develop an IT transformation strategy based on SOA. The organisation wants to increase its operational efficiency, introduce new products and to expand internationally. CSC is developing a services approach that leverages their current investment in large ERP systems and other legacy systems to support an operating model that is customer centric, employs standardised processes and provides the agility that will allow the organisation to be a product leader.

#### \* Real Time Responsiveness

SOA technologies are geared towards capturing and processing real time events, supporting the use of sense and respond technologies such as Radio Frequency Identification (RFID) and Business Activity Monitoring (BAM). If business services are designed to encapsulate business processes, responding to a business event can be as simple as invoking a business service.

### CASE STUDY

Several years ago a large government organisation wanted to improve the flexibility and agility of their systems to cater for changes in legislation and changes in government policy. They engaged CSC to develop a solution and, as a result, have a mature application architecture based on business services. Today, they are implementing changes in their systems to support complex changes to legislation in a fraction of the time that it used to take and are reducing the risk of introducing such changes by delivering them in manageable chunks. CSC still plays a role in architectural governance and in ensuring that the architecture is kept in line with the business direction.

#### \* Channel Independence

Business services lend themselves to re-use across channels, allowing organisations to add new communication and payment channels for customers and suppliers easily and quickly, without having to adapt applications. This means, for example, that it does not matter which channel a payment arrives on, with SOA it can be processed by the payment service.

### CASE STUDY

A large securities broker wanted to increase its ability to handle new channels. CSC used SOA to decouple the back-end applications from the user interface such that user interface development could be outsourced to a third party - allowing the company to concentrate on developing their back-end services. As a result the organisation's major business services can be rolled out to a wider cross section of clients giving them competitive advantage and reducing system development costs.

#### \* Reduced Point-to-Point Spaghetti

The point-to-point integration phenomenon is well documented, the bottom line being that if an organisation has  $n$  applications that all want to talk to each other, the point to point approach results in the construction of  $n(n-1)$  interfaces. Under this regime, 20 applications will require 380 interfaces. In large organisations this leads to a high degree of complexity that not only makes maintenance a nightmare, but makes changes complex and slow.

Using an SOA architecture approach with a services model significantly reduces the spaghetti, taking organisations close to the ideal of  $n$  interfaces for  $n$  applications - reducing the 380 interfaces in the previous example to close to 20.





## Which organisations will benefit most from SOA?

Organisations that benefit most from moving to an SOA model have large and complex application portfolios with a plethora of point-to-point interfaces. This is because the more complex applications and their integration architectures become, the more risky it is to change them.

With a high level of complexity, you eventually reach a point where you cannot accurately assess the impact of change, so test cycles become longer and more defects find their way into the production environment. When this happens, system change cannot keep pace with business change and the organisation loses its competitive edge: the

business cannot get to market on time or respond quickly enough to changes in demand.

A common symptom of organisations with this level of complexity is where a myriad of small database systems (eg: Microsoft Access systems) have sprung up to cater for the gap between what the business applications can support and what the business needs. Apart from providing only short-term respite from the problems of complexity, this type of bandaid solution is extremely damaging: reducing your control of corporate data; increasing your security and privacy risks; and creating expensive inefficiencies through duplication.



### CASE STUDY

A large government department wanted to replace a large and strategically important mainframe application with something that would support the new way in which it wanted to do business. By developing a program with them to replace the application piece by piece with services, CSC has helped manage the risk of interruption to the business inherent in replacing such a large system. By developing a roadmap for migrating major business systems to a services model at an enterprise level, CSC has provided a framework for future system development and acquisition that aligns technology with business imperatives.

#### Streamlined Costs

Evolving an SOA across the enterprise frees up IT resources and helps to ensure that your IT investments are focused on core capabilities aimed at growing the business.

### CASE STUDY

A large commercial insurer wants to migrate their legacy systems, mainly developed in IMS and Smalltalk. CSC is assisting this organisation to create business applications centred on their key business capabilities, and in the process integrating applications and reducing costs.

#### \* Shorter Development Times

The major benefit of re-use is shorter system development times and hence substantially reduced development costs. Development times are also compressed because, while they must conform to a consistent architecture, self-contained services can be built

independently of each other increasing the potential for parallel development. This means they can have their own release cycles, removing the need for “big bang” application releases.

#### \* Reduced Duplication

Re-use also reduces duplication. Whereas previously organisations would be likely to have multiple applications performing the same functions in slightly different ways, this is no longer necessary under an SOA.

#### \* Lower Cost of Support and Maintenance

Maintaining a sub-process as a contained service simplifies the task of maintenance. Changes can be made once and in one place for all business applications that use the service, reducing the testing scope. Composing applications from services means that, while a service must still be rigorously tested, the amount of testing required for each application that uses a service is substantially less.

### CASE STUDY

A large government utility wanted to streamline its business processes by improving integration with other government organisations and expanding its channels for doing business. CSC developed an SOA solution that reduced manual handoff to other organisations and established re-usable services that can be accessed via an internet portal or directly by its customer’s applications. The result of re-use of business services and reduced manual intervention in business processes has reduced costs.

#### \* Maximising ROI on Legacy Investments

As well as providing a level of re-use, wrapping a legacy application with a services interface can prolong its life, allowing you to maximise your investment in legacy business applications you cannot afford to throw away.

Of course SOA is not a total panacea. Legacy systems often reach a state where they are complex and difficult to change. Putting a standard services wrapper on them will not make them any easier to maintain but it will make them accessible to a wider range of applications.

At the same time, centralising support of the SOA technologies that comprise the services infrastructure maximises the return on investment in the skill set required to maintain the infrastructure.

#### CASE STUDY

A large manufacturing company wanted to modernise its functionally effective but aging suite of business applications. The solution that CSC developed with the organisation is based on SOA and uses agile, best of breed software in concert with large ERP modules while retaining the aspects of the legacy systems that contributed towards the organisation's competitive advantage. The organisation has taken the opportunity to restructure their business processes to remove duplication and inefficiency. The SOA approach allows them to implement improved business processes incrementally, minimising risk to business continuity.

### Recognising the Stumbling Blocks

Organisations trying to implement SOA typically fall down in three major areas.

#### 1. Failure to build a business case for SOA

IT departments sometimes make the mistake of treating the introduction of SOA as a purely IT initiative. Without engaging the business and establishing the business benefit of SOA it will be hard to justify spending the amount of money necessary to make it successful. If the business perceives the introduction of SOA as an IT exercise with no business benefit, they will squeeze the IT department to keep costs as low as possible from the outset. Organisations are no longer interested in investing large sums of money in technology for technology's sake. An ill-conceived SOA implementation runs the risk of only adding to the confusion of technology already existing within the organisation.

#### 2. Failure to take a holistic architectural view of SOA

Sometimes organisations believe they can buy an SOA from a vendor and bypass the normal architecture definition activities. Technology is an important part of SOA but successful implementation also requires the business to consider areas of organisation, business location, business

process, corporate data design and business application design. In particular, it is vital to consider the impact of introducing a 'services based' application model on the organisation both from the view of maintaining and developing the applications and operating the business.

#### 3. Failure to adequately plan the introduction of SOA

Introducing SOA into a large organisation presents many challenges. Planning its introduction will enable you to minimise risk to the continued operation of the business and realise early benefits. Technology vendors often fail to draw attention to the importance of such planning and give the impression that introducing SOA is a simple matter of installing the enabling technology.

### Avoiding the Pitfalls

Introducing SOA is not easy but it is not impossible either. Following are some practical ways of maximising your chances of a successful SOA implementation.

- **Focus on delivering early business benefit**

SOA must be introduced incrementally keeping in mind optimal business returns at all times. Identify manageable pieces of work that can be delivered as a series of small successes that will deliver a measurable business benefit. Consult business groups when deciding which applications to attack first. Pick a business pain point, not just something that will be easy to implement. Design the architecture to support incremental implementation where possible.

Interdependencies between projects implementing the roadmap must be identified and managed. Priorities must be determined for projects that compete for resources so that business objectives are met. Timing of solution delivery must provide downstream projects with the necessary pieces of the solution to build on. Incrementally rebuilding strategic business applications to conform to the new model represents a potential risk to business continuity and that risk must be managed.





- **Get your business units to collaborate**

Your business units must agree on how they will perform “common” business tasks. Application development groups must adopt a collaborative approach both with the business and with each other. They must foster a “re-use” rather than “cut and paste” culture.

- **Align the technology with the business**

SOA technology vendors are only too willing to demonstrate how their products can be applied to the organisation’s business problems. Sometimes vendors create the impression that, because the technology can solve a business problem in the demonstration environment, it can do the same in the business environment without significant effort.

Do not be too concerned with technology too early in the process. Worry about defining services first then decide on the technology to implement them, based on the business and technical requirements. Remember that technology is an enabler; it is not architecture in and of itself.

- **Favour buy over build**

There is a raft of package software available for constructing the plumbing that enables applications and services to work together to implement business processes. These range in sophistication from message queuing software to enterprise service software to products that purport to provide a complete “end to end” technical integration solution. Select the appropriate technology based on support for open standards, ability to meet performance and availability requirements, ability to meet connectivity requirements and total cost of ownership. It is generally considered good practice to implement the business requirements of a service using package software also. Package software generally provides the most cost efficient solution and the leading vendors stay in step with industry best practice.

- **Future Proof**

Many of the major vendors such as Oracle, IBM, webMethods, SeeBeyond and even SAP are touting their products as SOA enabled or web services compliant. A major consideration when selecting technology should be the strength of the commitment on the part of the vendor to supporting open standards.

Adoption of open standards presents the best chance of avoiding vendor lock in and adding yet another piece of legacy technology to what has become an ever expanding portfolio for many organisations. The problem is that there are so many standards to choose from.

As one ascends through the technology stack their number increases and the degree to which they provide true standardisation decreases. The best way to minimise the risk of technology obsolescence at the upper levels of the technology



stack is to select products comprised of components with clearly defined interfaces. This provides opportunity for their reasonably painless replacement at a later time, if necessary.

- **Put governance processes in place**

Manage the expectations of stakeholders. Put governance processes in place to ensure architectural components are built according to the architecture strategy. This will ensure you develop services that both serve the architectural model and deliver business value. Architecture governance identifies standards and approaches to be followed when implementing the architecture. It also specifies how stakeholder groups are catered for and how priorities are set. The Architecture governance strategy defines roles and responsibilities necessary for a collaborative approach.

- **Address the organisational issues early**

To be truly effective, enterprise architectures must consider people. There may be a shift in organisational structure required to accommodate centralised administration of skills, priorities, technologies and standards. There may be additional organisational structures required to introduce governance for process maturity strategy, architecture strategy and architecture governance strategy. A shift in the mindset of development teams and business units to a more open and co-operative mode of working will almost certainly be required. It will be necessary to abandon the ‘cut and paste’ mindset in favour of the ‘reuse’ mindset. Initially, it may even be necessary to introduce a reward system to foster such a change of mindset until the benefits become tangible.



- **Decide how much work a business service should do**  
Getting the level of granularity of the building blocks right is crucial to getting return from re-use and minimising the impact of change. This means correctly deciding the amount of work that a service will perform. Generally speaking, granularity should be at the level of a business sub-process. Services designed at the correct level of granularity will contribute towards containing the impact where change is required. The impact of a change to a sub-process encapsulated within a service will be restricted to the service itself and will not be felt by the applications that use the service. For example, if a change is made to a payment sub-process that is encapsulated in a service, the change should be transparent to the applications that use the payment service, restricting the ripple effect of change.
- **Make each service a black box**  
It should be possible to change the workings of a service with minimal or no impact on its clients. The interface is the only part of the service that a service client need know about. This includes any input parameters, pre-conditions, output and post-conditions. For example, if a service makes changes to the database, potentially affecting how other applications or services work, then those changes represent part of the service interface.
- **Make reliability a priority**  
Getting the implementation model right to achieve the required level of performance is crucial. If the service cannot be trusted to achieve its performance and reliability targets nobody will use it.
- **Periodically review the architecture.**  
Review your architecture strategy periodically to ensure it is still aligned with your business vision. Be prepared to re-factor parts of the architecture as the result of lessons learned from implementing it. Services must be periodically reappraised as to whether they are at a level of granularity that provides the best re-use, flexibility and maintainability.

## References

*Service Oriented Architecture - Developing the Enterprise Roadmap, the Burton Group, February 2004.*

*Nine Tips for SOA Implementation, Forrester, April 2004.*

*Migrating to a Service-Oriented Architecture - Part 1, IBM, December 2003.*

*CSC e4sm Conceptual Reference Architecture, Version 2, November 2004.*

[www.ebpm.org](http://www.ebpm.org)

## Acknowledgements

*The author wishes to thank CSC colleagues, Geoff Brehaut, Bill Bolton, Graham Mackenzie, Craig Perritt and Matthew Dawson, for their valuable contributions to this White Paper.*

## Conclusion

Organisations cannot realise the benefits of SOA by buying technology off the shelf. A successful SOA implementation requires the holistic treatment of business process, organisational issues, business locations, applications, data and technology.

However, properly planned, managed and rolled out, an SOA can deliver an architecture that both streamlines IT costs and facilitates true organisational agility.

## **Computer Sciences Corporation**

### **CSC Australia Pty Ltd**

**ABN 18 008 476 944**

For more information contact:

Tel: 1300 551 272

<http://www.csc.com/au>

## **Worldwide CSC Headquarters**

### **The Americas**

2100 East Grand Avenue

El Segundo, California 90245

United States

+1.310.615.0311

### **Europe, Middle East, Africa**

Royal Pavilion

Wellesley Road

Aldershot, Hampshire GU11 1PZ

United Kingdom

+44(0)1252.534000

### **Australia**

26 Talavera Road

Macquarie Park, NSW 2113

Australia

+61(0)29034.3000

### **Asia**

139 Cecil Street

#08-00 Cecil House

Singapore 069539

Republic of Singapore

+65.6221.9095

## **About CSC**

*Computer Sciences Corporation helps clients achieve strategic goals and profit from the use of information technology.*

*With the broadest range of capabilities, CSC offers clients the solutions they need to manage complexity, focus on core businesses, collaborate with partners and clients, and improve operations.*

*CSC makes a special point of understanding its clients and provides experts with real-world experience to work with them. CSC is vendor-independent, delivering solutions that best meet each client's unique requirements.*

*For more than 40 years, clients in industries and governments worldwide have trusted CSC with their business process and information systems outsourcing, systems integration and consulting needs.*

*The company trades on the New York Stock Exchange under the symbol "CSC."*