

SOA + OSS + Offshore Outsourcing = Success
A Torry Harris Article

INTRODUCTION	3
AGILE & INEXPENSIVE: SUCCESS FACTORS	4
AGILITY: REDUCED TIME TO MARKET	4
INEXPENSIVE: REDUCED OPERATIONAL COST	4
THE TRIO	6
SOA	6
OSS	6
OFFSHORE OUTSOURCING	7
SERVICE-ORIENTED ARCHITECTURE - A SAMPLE	8
SERVICE ACCESS LAYER	8
SERVICE LAYER	9
<i>Orchestration Service Layer</i>	9
<i>Application Service Layer</i>	9
INTEGRATION LAYER	9
APPLICATION LAYER	9
PERSISTENCE LAYER	10
OPERATING SYSTEM	10
SERVICE RUNTIME	10
<i>Service Registry</i>	10
<i>Quality of Service</i>	10
<i>Service Deployment</i>	10
SOA IMPLEMENTATION	11
SOA & OSS	11
SOA & OFFSHORE OUTSOURCING	14
<i>Implementation Strategy</i>	14
<i>Incremental Approach</i>	14
<i>The Challenges</i>	15
<i>Offshore Advantage</i>	16
CONCLUSION	16

Introduction

Adopting SOA for the IT infrastructure is a significant paradigm shift. As per the leading market research firms, SOA is being adopted as the chosen computing methodology by several enterprises. The adoption rate is exponential.

The adoption within an organisation however needs to be incremental and evolutionary. Open Source Systems (OSS) by virtue of their low TCO (total cost of ownership), compliance to standards and being vendor neutral, allow an organisation to take the first step into SOA, all the way to a full-fledged SOA infrastructure in a planned and phased manner.

Implementation of SOA through OSS is best done through offshore outsourcing. Apart from cost savings, offshore outsourcing offers flexibility and scalability with respect to IT resources and skills. With compliance to CMM (Capability Maturity Model) and Information Security Bodies, offshore outsourcing providers offer high quality services at a lower cost.

The two primary business drivers of today's successful organisation are - 'Reduced Time to Market' & 'Reduced Operational Costs'. SOA, OSS and Offshore Outsourcing together address these drivers and improve the quality of service provided to the end customer.

This article gives an overview of these components and the options available to an organisation for implementing SOA through OSS and Offshore outsourcing. The target audience are organisations whose core business is not IT.

Agile & Inexpensive: Success Factors

Today's market conditions are more dynamic when compared to the previous century. The business needs to be agile to address the market conditions and be able to collaborate with other businesses (internal/external) to offer value added extensive services/products. All this has to be achieved at a lower cost, without impacting the quality of service provided. Inexpensive agility in IT infrastructure is one of the key success factors for an organisation.

Agility: Reduced Time to Market

Consumers of today want the ability to specify the options they want in services and products. Many market segments are close to saturation. The key differentiating factor between competitors is the 'effectiveness in meeting the customer expectations'. With today's dynamic market conditions, the IT infrastructure (including the IT Team) of a successful organisation should have the agility to react quickly to the changes in the market.

Inexpensive: Reduced Operational Cost

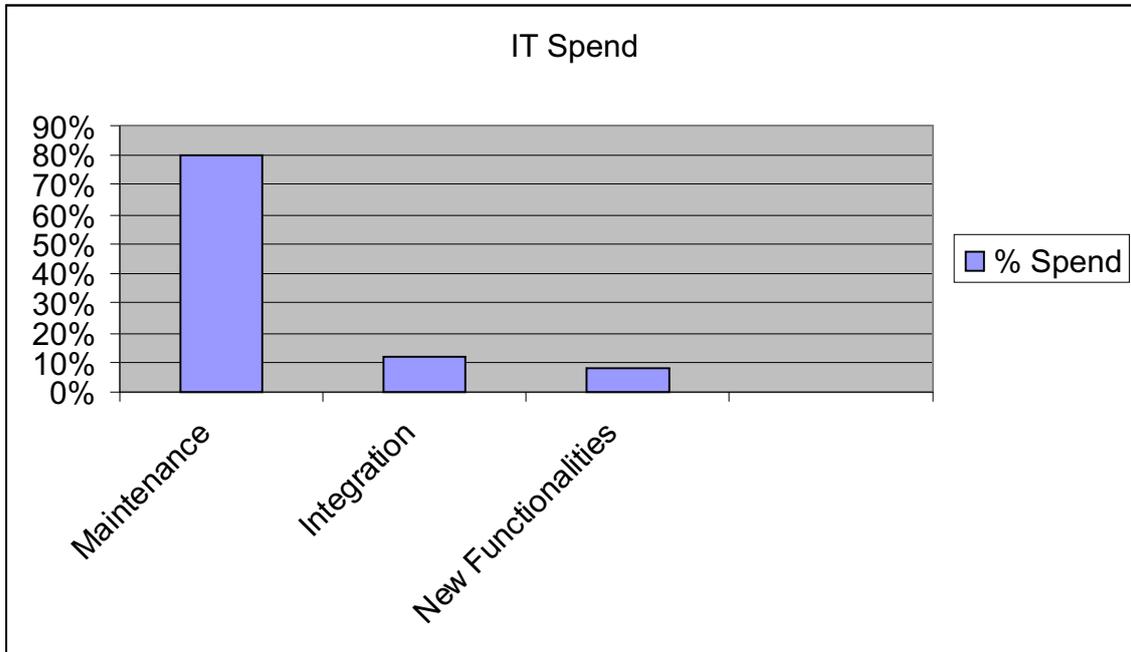
The following key factors can contribute to the increased maintenance costs of IT infrastructures:

- 1.) Software Product Licensing & Support
- 2.) System Implementation
- 3.) New Systems
- 4.) Integration
- 5.) Maintenance of existing systems (including enhancements & change requests)
- 6.) IT Resource Management Overhead
- 7.) Flexibility & Scalability in skills and numbers

With the cost of hardware reducing drastically, it is no more as significant an impact on a typical IT Budget as it used to be.

According to a study conducted by a leading market research firm, 80% of typical IT budgets are earmarked for maintenance, 12% for integration and the remaining 8% for implementing new functionalities. The higher cost of system maintenance and integration are primarily due to the,

- 1.) Maintenance of disparate monolithic systems
- 2.) Integration of these systems with newer technologies and
- 3.) IT Resource Requirements/Management required for maintenance and integration
- 4.) Non-compliance to standards (proprietary in some form) resulting in ineffectively leveraging the capacity/ability of these monolithic systems



These factors, added to the IT Resource Management Overhead can severely impact an organisation's budget and eventually, its competitiveness.

The organisation of today has to run a very cost effective IT Infrastructure by leveraging and reusing existing infrastructure, implementing low cost and high value IT systems that comply to standards.

The Trio

One of the ways to achieve the success factors described above is through a combination of SOA, OSS (Open Source Software) and Offshore Outsourcing. While SOA provides the agile & reusable architecture, OSS provides the tools at a lower TCO and Offshore Outsourcing helps in a cost effective implementation and maintenance, at a lower IT resource management overhead.

SOA

Service-oriented architecture (SOA) is the technical enabler for the business of today - to efficiently collaborate with other businesses/divisions and react quickly to the ever-changing market conditions at a reduced cost.

SOA is neither a technology nor a product, it is a methodology. It is a computing methodology that provides guidelines to architect your IT environment to react quickly and cost effectively to the business changes.

OSS

Open Source Software (OSS) refers to the community of free applications and systems being written by developers around the globe. Open Source Software is freely distributable with access to the source code of the software.

The number of OSS applications available in the market today is an example to show the increasing demand by consumers. Apart from the fact that OSS is free; it is widely accepted by enterprises and major product vendors because of the demand for constant improvement of their IT environments, innovation to keep up with the business changes and supreme quality. These products go through several test cycles for different scenarios by IT personnel worldwide, thereby creating a very robust system.

Apart from the support provided by Open Source forums, several IT Service Providers including the major product vendors, also offer support to a wide range of OSS.

There are several licensing terms under which OSS is distributed. The most commonly used Open Source Licenses are

License Type	Website
Apache License	www.apache.org/licenses
BSD	www.opensource.org/licenses/bsd-license.php
GPL	www.gnu.org/licenses/gpl.html
LGPL	www.gnu.org/licenses/lgpl.html

Details of the other Open Source licensing options can be found at, www.opensource.org/licenses

Offshore Outsourcing

Offshore Outsourcing is of increasing importance to organisations seeking to reduce IT expenditure. Significant cost reduction, which can be achieved in a planned, phased manner, is just one of the several benefits of offshore outsourcing. It provides the flexibility and agility required by the business of today to react to changing market conditions.

The offshore team is a virtual extension of the outsourcing organisation's IT team. Given the advantage of the time-zone difference, a 'follow the sun' development model can be adopted. This model with the appropriate use of onsite and offshore resources, will enable the organisation's IT team to have extended work days upto double the normal business hours.

With the availability of a wide range of technical skills, offshore outsourcing increases the potential of an organisation's IT team exponentially. The IT team now will have access to the varied skills and the capability to scale on demand, without the overhead of resource management that involves recruitment and training.

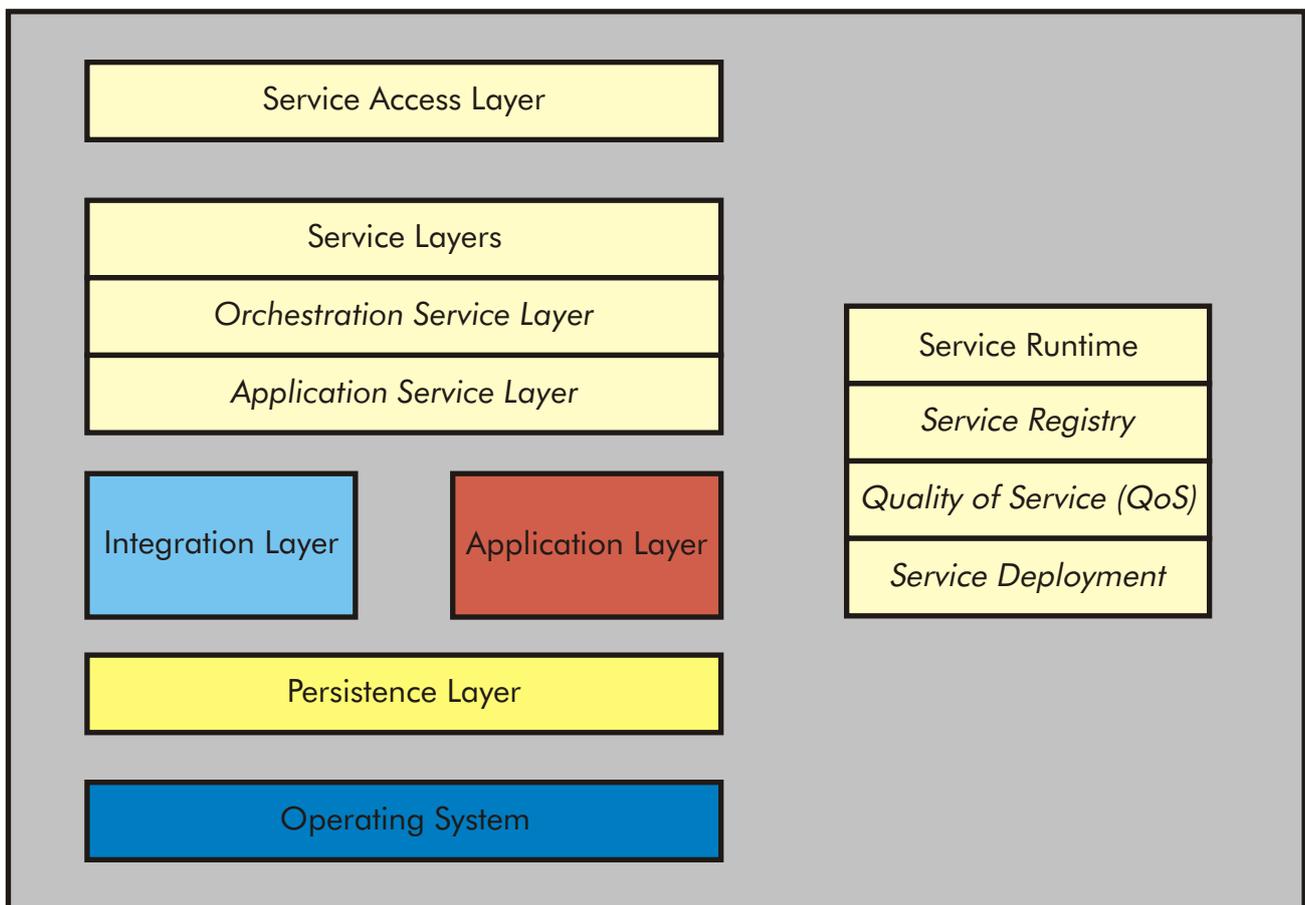
Summarizing the key benefits, Offshore Outsourcing offers,

- 1.) Cost Savings
- 2.) Increased Quality of Service
- 3.) Flexibility IT Skills
- 4.) Scaling on demand
- 5.) Extended work hours

Service-oriented architecture - a sample

Service-oriented concepts do not impose any default architecture. However, they do define the characteristics of the elements that comprise the target Service-oriented architecture. This section describes a sample architecture based on Service-oriented concepts. In the sections below, the elements in this sample architecture are mapped to suggested OSS tools/products.

The architecture and the listed OSS tools/products are based on the assumption that 'Web services' are used for the implementation of the Service-oriented architecture. The 'services' refer to 'Web service' in this context.



Service Access Layer

The Service Access Layer is the window to an SOA based IT System - it is the interface layer between the end client and the system. The client list includes human or application interfaces (from external systems) through a variety of channels including web portals, handheld devices, etc.

Service Layer

The Service Layer forms the crux of the Service-oriented architecture - the services. This layer defines and exposes the business and application functionalities supported by the system as services, to both internal and external systems. Services are composed and orchestrated to execute a defined business process in this layer. Based on the abstraction and the purpose, this layer can be further sub-divided into,

- 1.) Orchestration Service Layer
- 2.) Application Service Layer

Orchestration Service Layer

This layer is fundamental in realising some of the key benefits of SOA - composability, reusability and abstraction. The services in this layer define and expose the business processes of the system. The services can be either coarse or fine grained. If coarse grained, then typically, they are composed of two or more fine grained services.

Business Process Execution Language for Web services (BPEL-WS) is the standard specification for orchestrating web services based on a business process.

Application Service Layer

The services in this layer are the 'workers'. They represent the actual application logic required to implement the business processes. These services are heavily reused by the services in the Orchestration service layer to execute a business process.

Apart from core application logic, these services also offer utilities (like audit trails) and integration functionalities.

Integration Layer

This layer integrates the Service-oriented system to the existing systems. This is of importance when existing systems cannot be exposed as services that can interoperate directly with the Services-oriented system. The inability to expose the existing systems as interoperable services can be due to technical or performance (non-functional) constraints.

Application Layer

This layer hosts the implementations of the services exposed in the Application Service Layer. If a 'Bottom-Up' approach is used in implementing the Service-oriented system, then this layer will include both existing and new applications else if a 'Top-Down' approach is used, then it will have only new applications.

Persistence Layer

Typically this is the database layer, where the functional and non-functional data are persisted either temporarily (cache) or permanently (duration determined as required by the business and technical requirements).

Operating System

This is the operating system on which the various stacks of the Service-oriented system are hosted.

Service Runtime

This stack represents the components required to deploy and 'run' the services. The core components in this stack are,

- 1.) Service Registry
- 2.) Quality of Service
- 3.) Service Deployment

Service Registry

The registry is the central repository where all services are defined and exposed using standard specifications, to be discovered by both internal and external systems.

Quality of Service

For a given service, this component represents the provision of secure and reliable transaction capabilities at optimal performance levels through WS-* extensions.

Service Deployment

This is the platform for deploying services. This platform has to support the features of a Service-oriented architecture. The location of the deployment is abstracted from the Service Access Layer through the use of the Service Registry.

SOA Implementation

SOA, through OSS and offshore outsourcing is a high quality, low cost implementation of a business friendly architecture. This section details the various Open Source Software options available to implement an SOA and how offshore outsourcing addresses the challenges of an SOA Implementation.

SOA & OSS

The following table maps the various layers of a Service-oriented architecture described in the previous section to the available Open Source Software.

SOA Component	Open Source Systems	Description
Service Access	Apache HTTP Server	<p>Apache HTTP Server is a commercial grade web server. This is from the Apache Software Foundation.</p> <p>Website: httpd.apache.org License Type: Apache License</p>
Orchestration Service Layer	jBPM	<p>jBPM from JBoss enables the creation of business processes that coordinate between people, applications and services.</p> <p>Website: www.jboss.com/products/jbpm License Type: LGPL</p>
	ActiveBPEL	<p>The ActiveBPEL engine is a robust runtime environment that is capable of executing process definitions created to the Business Process Execution Language for Web services (BPEL4WS, or just BPEL) 1.1 specifications.</p> <p>Website: www.activebpel.org License Type: GPL</p>
Application Service Layer	Apache Axis & Apache Axis2	<p>Axis2 is a complete rewrite of the Apache Axis. Axis is an implementation of SOAP and allows development and deployment of web services (through a J2EE application server)</p> <p>Axis is an integral component of several Open Source and Commercial application servers.</p> <p>Website 1: ws.apache.org/axis Website 2: ws.apache.org/axis2 License Type: Apache License</p>

SOA Component	Open Source Systems	Description
Service Registry	jUDDI	<p>jUDDI is a Java implementation of the UDDI specification for Web services. It's an Apache project.</p> <p>Website: ws.apache.org/juddi License Type: Apache License</p>
Service Deployment	FUSE	<p>LogicBlaze's FUSE is a complete SOA platform. The components include a messaging platform, enterprise service bus, and required runtime service providers for routing, orchestration, transformations, security and service lookups. It can either be deployed on an application server or deployed to a network of brokers.</p> <p>Website: www.logicblaze.com License Type: Apache License</p>
	JOnAS	<p>Java Open Source Application Server (JOnAS) is a J2EE Application Server with support for Web services through Apache Axis.</p> <p>Website: jonas.objectweb.org License Type: LGPL</p>
	Tungsten	<p>Tungsten from WS02 is an application server for Web services with a built in web server and supporting WSDL and the various WS-* extensions. It's based on Apache Axis and can be hosted on a J2EE application server.</p> <p>Website: www.wso2.com/products/tungsten License Type: Apache License</p>
	JBoss Application Server	<p>JBoss Application Server is a J2EE application server capable of hosting Web services</p> <p>Website: www.jboss.com/products/jbossas License Type: LGPL</p>
Integration Layer	ServiceMix	<p>ServiceMix is a lightweight ESB implementation available as an Apache Incubator project. It's based on the Java Business Integration (JBI) specification (JSR 208). ServiceMix is implemented as a core component in other ESB products (details below)</p> <p>Website: servicemix.org/site/home.html</p>

SOA Component	Open Source Systems	Description
Service Deployment	FUSE	LogicBlaze's FUSE is a complete SOA platform. Among the other features, it also includes an ESB implementation based on ServiceMix Website: www.logicblaze.com
	Celtix	Celtix delivers a Java enterprise service bus (ESB) runtime and set of extensibility APIs, simplifying the construction, integration and flexible reuse of technical and business components using a standards-based, service-oriented architecture. Website: celtix.objectweb.org
	Mule	Mule is an Enterprise Service Bus (ESB) messaging framework with the core container based on SEDA (staged-event driven architecture). It is a scalable, highly distributable object broker that can seamlessly handle interactions with services and applications using disparate transport and messaging technologies. Unlike other JBI implementations, Mule is message type agnostic. Website: mule.codehaus.org
Application Layer	The Application Servers specified against 'Application Service Layer' can host the core applications as components of Java EE.	
Persistence Layer	Derby	Apache Derby is a relational database implemented entirely in Java. It has a small footprint of about 2MB for the base engine and the driver. Website: db.apache.org/derby License Type: Apache License
	PostgreSQL	PostgreSQL is a full-fledged relational database system with an unlimited database size. Website: www.postgresql.org License Type: BSD
	MySQL Community Edition	MySQL provides database products to meet the needs of ISV/OEM, Enterprise, and Community Users. The community edition is open sourced. Website: www.mysql.com License Type: GPL

SOA Component	Open Source Systems	Description
	Hibernate	Hibernate from JBoss is a object/relational persistence and query service. It offers both first (transaction level) and second level (class or collection level) caching features Website: www.hibernate.org License Type: LGPL
Operating System	Linux	Linux is the most popular Open Source Software. Linux a flavour of UNIX is available as several distributions (supported and unsupported). Red Hat and Mandrake are two distributions. Website: www.linux.org License Type: GPL

SOA & Offshore Outsourcing

Implementing SOA in an organisation can be a long-drawn process. With an incremental approach, realisation of the business benefits can be achieved in a phased manner. The following sections detail the incremental approach, SOA Implementation challenges and how Offshore Outsourcing addresses these challenges.

The Implementation Strategy

There are a few standard strategies such as 'top down', 'bottom up', and 'meeting in the middle' that are used for an SOA implementation. The choice is dependent on the specific requirements of an organisation wishing to adopt SOA.

Top Down

'Top Down' is the ideal strategy, a near utopian solution. This is normally adopted by an organisation (or a division) that is building its IT infrastructure ground up for the first time. Thorough and complete analysis of the business requirements precedes identification of Services and implementation of the new system/s based on Service-oriented concepts.

Bottom Up

'Bottom Up' is the most practical strategy, where the Service-oriented architecture is implemented on the foundation of an existing IT infrastructure. This allows maximum reuse. The implementations are specific to the current business requirements. There is a down side to this strategy - the probability to deviate from Service-oriented concepts is high owing to no initial analysis of the entire IT landscape.

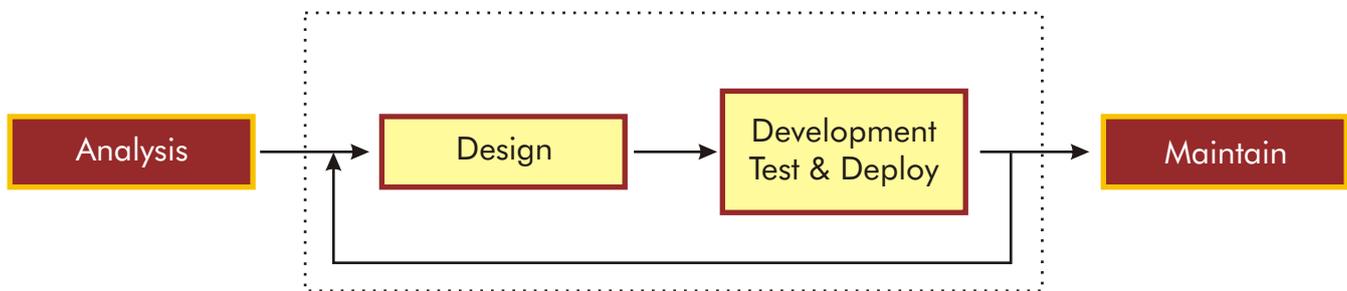
Meeting in the Middle

The optimal strategy is 'meeting in the middle'. This advocates analysis from a long-term perspective and parallel implementation. It reuses the existing infrastructure to deliver "today's" business needs.

Incremental Approach

Be it 'top-down', 'bottom-up' or a combination of the two, any SOA strategy implemented in an incremental approach helps the organisation to achieve business benefits.

An incremental approach involves design, development, testing & deployment of services based on business priorities. The process is incremental and begins with an analysis phase, which identifies the service candidates in line with the business priorities. At the end of each cycle, the services implemented enter into a maintenance phase for ongoing support and enhancements.



The Challenges

Though an incremental approach assists in realising business benefits, there still remain a few challenges that need to be addressed:

1.) Flexibility

The resource profiles and bandwidth required in the various phases of an SOA Implementation would vary significantly. For example, within one incremental SOA cycle, the resource requirements (including profiles) vary from design to development to the testing phases. Further, depending on the scope of a particular incremental cycle, there could be significant variations in resource requirements compared to the previous or subsequent cycles.

2.) Skills & Scalability

The choice of technology to implement an SOA depends on several factors including cost, existing IT infrastructure, product features, etc. Expertise in new technologies/products is now required almost overnight. As SOA is adopted across various divisions of an organisation, the resource availability would also need to scale up as per the new skill requirements. Along with resource flexibility, this poses another resource management challenge.

3.) Throughput

SOA Implementation is an organisation wide programme & not a one-off project. There is scope for decrease in the average delivery throughput as each incremental cycle progresses.

4.) Quality

The quality of the output could be severely impacted due to inflexibility in resourcing and lack of

in-house IT expertise. If appropriately skilled resources are not deployed for the relevant Incremental phase, there would be a Quality lapse.

5.) Cost

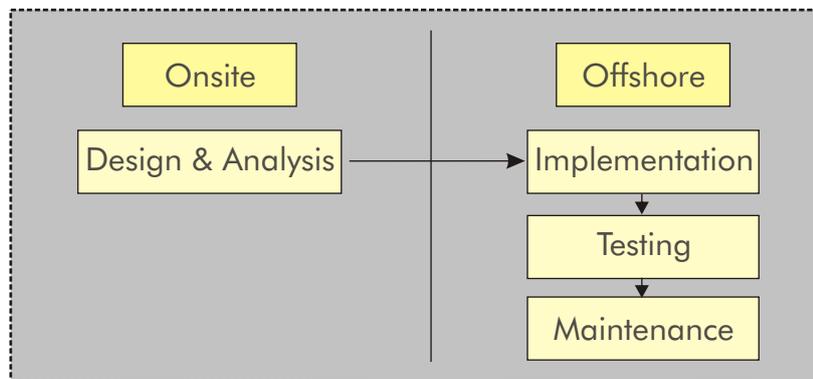
While Open Source Software will assist in reducing the TCO of the products used for an SOA, IT resources and resource management will still constitute a significant portion of the SOA budget.

Offshore Advantage

Offshore outsourcing is an ideal solution for an SOA Implementation. The challenges detailed above can be addressed by implementing an onsite/offshore resourcing model.

Onsite/offshore Model

In an onsite/offshore model, the core analysis/design team from the service provider will work onsite at the organisation's (customer) premises. The onsite team will interact with the business and IT teams of the organisation, analyse and identify the service candidates and design the framework and the services. The development and test phases of the Incremental approach will be taken up by the offshore team with support from the onsite team (for clarifications).



The offshore team will further provide support (maintenance) of the implemented services including any legacy/existing IT infrastructure.

With access to a wide range of high caliber skill pool and compliance to industry standards like CMM, ISO27001 (Information Security), the offshore service provider can offer a high quality service with the required flexibility and scalability of the resources, at an optimal cost.

Conclusion

Through Open Source Software and Offshore Outsourcing, an organisation can realise the benefits of SOA in a phased manner. A combination of SOA, OSS and Offshore Outsourcing will enable an organisation to be agile and maintain a low operational cost, which form the key success factors in today's dynamic market conditions.