

Agile Offshore Development with SOA





Table of Contents

| | |
|--|----|
| Introduction..... | 4 |
| Modern Offshore Development..... | 5 |
| Benefits..... | 5 |
| Risks..... | 5 |
| Agile and SOA..... | 6 |
| The Agile Development Process..... | 6 |
| Service Oriented Architecture..... | 8 |
| THBS Best Practices..... | 10 |
| Team Integration..... | 10 |
| Team Composition by Functionality..... | 10 |
| SOA-Driven Iteration Planning..... | 11 |
| SOA-based Continuous Integration..... | 11 |
| SOA and Test-Driven Development..... | 11 |
| Immediate Feedback on Functionality..... | 12 |
| SOA-Driven Documentation..... | 12 |
| Conclusion..... | 13 |
| References..... | 14 |
| Links..... | 14 |

abstract

Agile Development and Service-Oriented Architectures (SOA) will cause major shifts in development strategy, application acquisition and systems integration.

Global IT services and solutions providers like THBS bring Agile, SOA, and Offshoring together in order to address the primary need for today's business – the ability to adapt to changes quickly and cost-effectively.

This whitepaper presents a set of best practices around Agile Development and SOA which help customers to improve the ways how they can manage their offshore partner in order to improve cost efficiency, risk management and flexibility:

- ▶ Benefits and risks of modern offshore development
- ▶ Efficient strategies to mitigate key offshore risks
- ▶ Best practices of THBS to make offshore development more efficient

Introduction

It has been a long way from the tayloristic work models of the early 19th century to the advanced toyoistic work models of our time which include concepts such as Continuous Improvement, Kanban, and Just in Time Production (JIT).

In parallel, many industries have already undergone massive re-organizations and optimizations of their value chains.

It often seems as if the IT industry is still stuck in the early phases of these processes.

However, it is unquestionable that today's constant cost and efficiency pressure is also forcing our industry to look out for similar ways to improve our production methods and to optimize the IT value chain.

Offshore-development with the promise of cost efficiency and resource availability has been looked at as a way to optimize the IT value chain by outsourcing tasks like implementation and maintenance.

However, in the past many customers have looked at quasi-tayloristic production methods in offshore developments, often centered on waterfall-like development models.

Similar concepts could be found in other industries decades ago.

However, modern supply chains require a much higher degree in flexibility and efficiency.

Hence, many vendors have invented models which are better suited in such situations.

For example, many car manufacturers are now allowing their suppliers to put their own production facilities directly into the car manufacturer's factories.

This kind of tightly integrated production and supply chain is a prerequisite to make concepts such as Just in Time production possible.

In the same way opportunities lie ahead for business organizations and their IT.

Agile development methodologies and Service-Oriented Architectures (SOA) will cause major shifts in development strategy, application acquisition and systems integration.

This will affect how global IT services and solutions providers are able to contribute to the economic success of their business partners.

When brought together, Agile, SOA, and Offshoring will address the primary need for today's business – the ability to adapt to changes quickly and cost effectively.

This whitepaper presents a set of best practices around Agile Development and SOA which could help customers to improve ways in which they can manage their offshore partner in order to improve cost efficiency, risk management and flexibility.

Modern Offshore Development

Benefits

In the past, many customers looked at offshoring predominantly as a way of cutting costs.

While offshore locations are indeed providing highly qualified technicians and other specialists at low costs, one has to consider two key factors: the cost for developers and engineers are typically only a part of the overall project costs.

Furthermore, it must be seen that offshore development is increasing the complexity and hence the costs and risks of a development project.

It is important that the customer and the offshore provider agree on realistically achievable cost savings.

THBS often helps customers reduce their development costs by 20-30%.

Also, it is important that the customer and the offshore provider are jointly developing methods to address the pitfalls of offshore development, such as the ones presented in this whitepaper.

In addition to the cost benefit, many customers are today looking at offshore provider as a means to improve flexibility and in particular to overcome local shortages of skilled resources.

For example, if a customer is successful in outsourcing maintenance and support for a legacy application to an offshore location, this means that he can free up local resources for new projects.

Also, being able to tap into offshore resource pools means that a customer can more flexibly ramp projects up and down, allowing him to manage his project landscape in a more demand driven manner.

Risks

Of course, offshoring is not without risk, like any IT project. There have been many different statistics on the failure rate of IT projects in the past, some putting the number of failed projects at over 50% (see (Sta05)).

However realistic this number may be, the fact remains that IT is a risky and complex business, and adding offshoring to the equation is not reducing the complexity.

In a recent report, Gartner Group has identified the following key offshore risks (see (Gar)):

- ▶ Missed cost savings goals
- ▶ Quality problems
- ▶ Cultural differences and communication problems
- ▶ Legal issues
- ▶ Loss of control

We at THBS understand these risks, and we continually work together with our customers to ensure that these risks are controlled and managed in the best possible way.

This whitepaper outlines some efficient strategies to mitigate these risks efficiently.

Agile and SOA

The Agile Development Process

Agile Software Development methodologies emerged in the mid-1990ies as a counter-position to the traditional waterfall models, which dominated the IT world until then. The waterfall approach was increasingly seen as problematic, because it was largely seen as bureaucratic, slow, demeaning, and inconsistent with the ways how software developers should perform their work to be most efficient.

A key problem of the waterfall model is the inflexible division of a project into separate stages, which requires specifications to be complete and stable in an early stage, and causes problems in particular in situations where requirements are changing throughout the project or are not 100% clear in the beginning, which is almost always the case in complex IT projects.

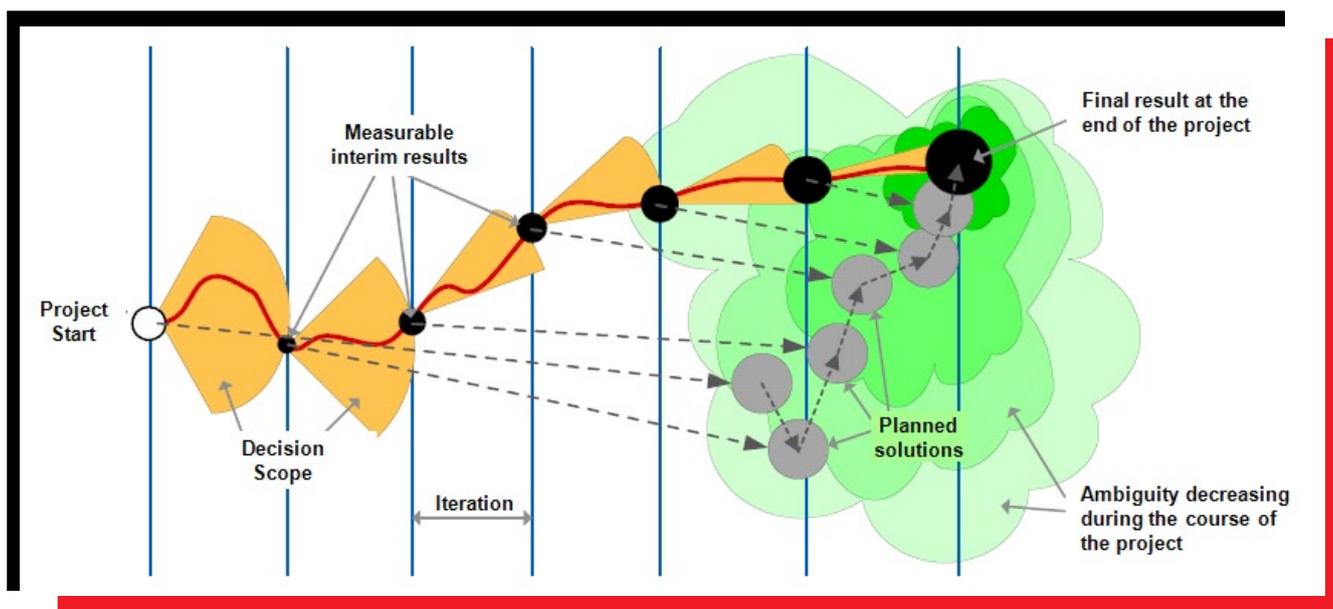


Figure 1: Agile, iterative development allows to deal with unclear requirements in the early phases of a project (© oose Innovative Informatik GmbH 2006)

Agile development helps minimize risk by developing software iterations, which typically lasts from two to four weeks.

Each iteration passes through a full software development cycle, including planning, requirements analysis, design, unit testing, and finally coding until the unit tests pass and a working product can be demonstrated to the stakeholders.

A key benefit of this is that software produced by the offshore developers can be demonstrated to the customer on a regular basis, dramatically increasing transparency and reducing the risk of projects going in the wrong direction.

In 2001, a number of prominent figures in the field of agile development created the Agile Manifesto, widely regarded as the canonical definition of agile development and accompanying agile principles. Some of the principles included are:

- ▶ Customer satisfaction by rapid, continuous delivery of useful software
- ▶ Working software is delivered frequently (weeks rather than months)
- ▶ Working software is the principal measure of progress
- ▶ Even late changes in requirements are welcome
- ▶ Close, daily cooperation between business people and developers

- ▶ Face-to-face conversation is the best form of communication (co-location)
- ▶ Projects are built around motivated individuals, who should be trusted
- ▶ Continuous attention to technical excellence and good design
- ▶ Simplicity
- ▶ Self-organizing teams
- ▶ Regular adaptation to changing circumstances

While many of these Agile principles can be applied in an Offshore development, one of the key requirements is impossible to achieve: co-location of the development teams.

Consequently, Agile Offshore development will have to develop solutions to address this issue.

THBS has developed a number of Best Practices which help address this issue, in order to make agile development feasible in a distributed offshore scenario.

Prior to this, we will take a quick look at the second key tool to modern offshore development, namely SOA.

Service-Oriented Architecture

An Enterprise SOA defines a business oriented view on the structural arrangement of individual application components, and how they interact, including basic application components, business processes and business rules.

In the last five years, the large software providers have created the necessary infrastructure for implementing an Enterprise SOA.

The table below describes the important elements of the Enterprise SOA.

| Enterprise SOA Element | Function | Examples |
|------------------------------------|---|---|
| SOA service | Stable, business-oriented basic application components. | Customer, loan application, contract, customer history. |
| Enterprise Service Bus (ESB) | Standardised communication infrastructure can be accessed via SOA services. | Query of the current processing status of the loan application. |
| Business Process Management (BPM) | Allows graphic definition of complex business processes by business analysts. Processes access basic SOA services during execution. | Process control of the loan application, including capture of application, credit check, credit score, decision, execution. |
| Business Activity Monitoring (BAM) | Enables monitoring of business processes. Important process KPIs are automatically captured and analysed. | Analysis of average processing costs and success in the risk management of loan applications. |
| Business Rule Management (BRM) | Facilitates control of business processes in the Enterprise SOA by means of business rules which can be adapted by business analysts. | Control of the loan decision process based on the adjustable parameters |

Table 1: Elements of the Enterprise SOA (Source: "Enterprise SOA", Prentice Hall 2004)

The purpose of a SOA service is the encapsulation of a relevant aspect of the business.

A service consists of several parts, including a number of interfaces (e.g. defined in WSDL), a service contract (an informal specification of the service that describes the purpose, functionality, constraints, and usage of the service), and the implementation.

A service owns its business logic and data.

Data is shared with other services only through the public interfaces of the service, never through direct database access.

A service can be compared to a rather large module or an application sub-system.

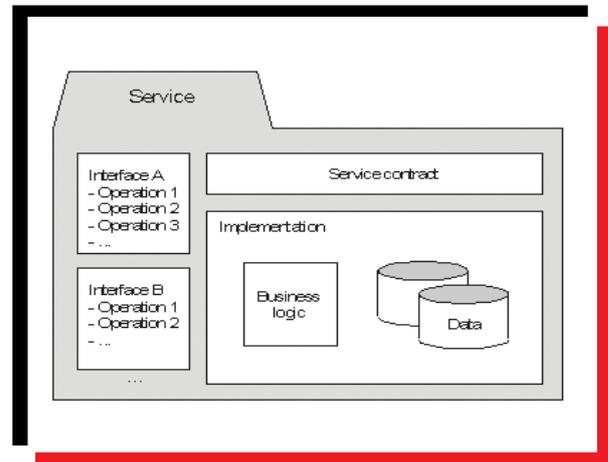


Figure 2: Definition of a Service (Source: "Enterprise SOA", Prentice Hall 2004)

An Enterprise SOA is typically comprised of a number of different services.

These services can be grouped into different layers.

Layering in an Enterprise SOA is independent of the underlying technology, and is more concerned with the lifecycle and the overall functionality of the service.

For example, basic services are typically relatively stable, i.e. they change less frequently, while process services are controlling business processes, e.g. through the use of BPMS and BRM, and are more frequently changing.

Mapping services to the right layer helps with more efficient management of the overall SOA landscape.

From an offshore development perspective, SOA is a powerful mechanism for managing the relationship between the customer and the offshore development team, as we will see in the following sections.

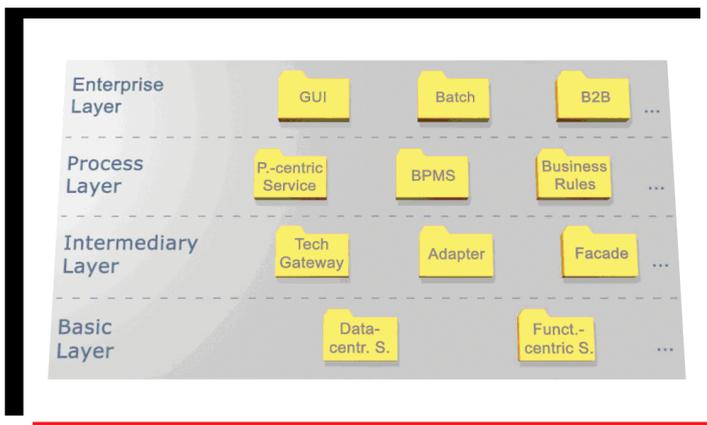


Figure 3: Enterprise SOA (Source: "Enterprise SOA", Prentice Hall 2004)

THBS Best Practices

The following section summarizes a set of best practices which THBS has developed in a number of projects, combining SOA and Agile to make offshore development more efficient.

Team Integration

Integrating the work across multi-site teams is challenging, especially if the teams are multi-national, multi-cultural, and are working across time zones.

The agile approach stresses the importance of face to face human interaction, which is difficult to achieve in a multi-site project.

In order to overcome this issue, THBS recommends that the following be implemented:

On-Site Teams: THBS usually sends a part of the team to the customer's location to work on-site directly with the customer.

Especially in the early phase of the project, this is an important measure to help ensure that key people in the project get to know each other personally and can establish a trust relationship.

Also, in the early phases, direct interactions help short communication cycles.

THBS can deploy as much as 30% of the overall team on-site for critical phases of the project.

Offshore Ambassadors: THBS usually recommends that the customer also sends representatives to work at the offshore location in India or China.

Such an ambassador is usually very well linked to the on-site team, and can thus leverage his contacts into the customer organization to help the offshore team to communicate more efficiently.

It often makes sense to combine a development ambassador with a business ambassador.

The business ambassador helps provide business context to the offshore team, which helps the offshore team to get a much better understanding of the overall project and the "why's" of many decisions.

Ambassadors should be rotated every few weeks or months, since to maintain contact with their home base, they can't fulfill their jobs anymore.

Address cultural issues: As an Asian offshore provider, THBS recognizes that it is absolutely vital for both sides to learn how to address differences in the culture.

To make Agile Development work, a lot of autonomy and decision making power has to rest with those team members who are actually responsible for the doing.

This is not exactly how a traditional "command and control" organization works.

THBS is constantly thriving to establish a culture which allows the individual to take responsibility for his/her own work in the Agile sense.

Establishing a personal relationship with the customer through on-site visits and the ambassador mechanism helps build a trust relationship, which is also important for offshore team members to overcome their own cultural barriers and work more effectively in a mixed environment.

Team Composition by Functionality

When applying the traditional waterfall model to offshore development, one often sees a tendency to define the onshore/offshore boundaries through the activities that people do.

Often, analysis and design is done onshore, implementation and unit testing offshore, and acceptance testing again onshore.

THBS has found that in order to work most efficiently, it often makes sense to draw the boundaries between onshore/offshore work not along the lines of activities, but rather along functionality.

Especially with SOA, it is possible to break up a system into loosely coupled services (each service comprises of a set of interfaces, business logic, and data).

The offshore team is often very well able to take full responsibility for the complete development cycle of some or even all services, allowing the onshore team to focus on other issues, such as the overall architecture, etc.

An important prerequisite for team composition by functionality is to ensure that the offshore team is sufficiently equipped with local analyst capacities. The ambassador mechanism described above can be very valuable here.

SOA-Driven Iteration Planning

In the Agile approach, a high emphasis is usually put on the iteration planning.

Ideally, the whole team should be involved in the iteration planning to ensure that everybody is clued in on the tasks for the next iteration. Because of the distributed nature of an offshore development, the iteration planning meeting must be carefully prepared, in order to reduce communication inefficiencies during the meeting, which is usually done over the phone.

Consequently, in close cooperation with the customer the THBS onshore team maps each feature in the upcoming iteration to the different elements of the SOA which are required to work together in order to support the feature.

This should be done in a work breakdown structure (WBS) which leverages SOA as a high level structure, and which is largely agreed upon between the onshore and the offshore team before the actual meeting.

SOA-based Continuous Integration

Even with the SOA-based approach, integration problems will never go away. While it is often easy to specify the signatures of the interfaces of an SOA service, it is usually very difficult to get the semantics agreed upon. Adding a service contract to the service definition helps, but it is not a silver bullet.

Consequently, introducing a Continuous Integration approach to the project can help speed up task execution significantly and at the same time ensure that all parts of the project are always closely aligned.

Daily checks and overnight builds, so that the code is ready for inspection by the onshore team in the morning, are one of the striking advantages of offshore development.

Proficient service providers use frameworks and tools like Cruise Control for this process.

With the rise of service oriented architectures another interesting question appears: *Should the Continuous Integration approach be applied to the entire code base or only on individual SOA services?*

Since each SOA service basically owns its own database schema and business logic, it is entirely possible to separate the code base of all services and integrate them only at runtime.

This can be a powerful mechanism to manage very large projects more efficiently.

Obviously, great care has to be taken to manage individual versions of services and their compatibility amongst each other, especially at the end of each iteration.

SOA and Test-Driven Development

The Agile approach puts a lot of emphasis on test driven development. The idea is that tests are written before the implementation and serve as a requirements definition.

With an offshore approach, it often makes sense that the onshore customer writes up requirements as a narrative to define a feature.

An offshore analyst and/or tester can then translate this narrative into a formal test script.

The onshore customer then reviews the test script. Inquiries appearing during these early stages of the development cycle are extremely helpful to improve the quality of the specification for the subsequent implementation, testing, and deployment phases.

Resources in terms of business analyst, architect and developer time invested here are invested wisely because their effort will pay off later in the development cycle.

Agile software development process models like Scrum support this test-driven approach.

SOA can contribute a great deal to this approach: since service consumer and service provider are separated through interfaces, their individual development cycle is de-coupled, as

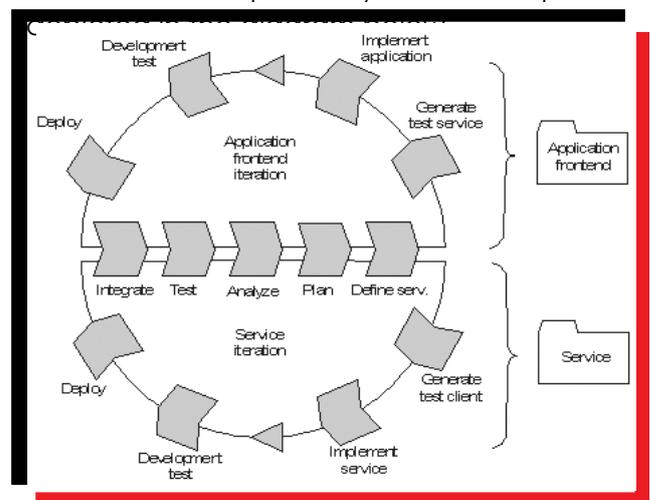


Figure 4: Individual development cycles of service consumer and service provider
(Source: "Enterprise SOA", Prentice Hall 2004)

THBS Best Practices

Modern SOA test tools allow complete simulation of service providers and service consumers. This allows for de-coupled development, but also for very efficient definition of test cases as part of a requirements definition.

Requirements can be defined, for example, as a combination of WSDL and test scripts.

An iteration plan can then define that in this particular iteration, a frontend should develop a certain functionality which will invoke a pre-defined WSDL interface in a pre-defined order, and react to pre-defined test data in a certain way.

Immediate Feedback on Functionality

In order to reduce the risk that the offshore team is not developing the functionality that was envisioned by the onshore customer, the previously described mechanism of Continuous Integration and SOA-driven test management enable THBS to provide customers with a concrete version of the software that can be tested at the end of each iteration.

This feature of Agile development is a key element to help support close alignment between the customer and the offshore team.

THBS encourages customer to participate in regular remote demos of the current state of the software through remote desktop software, presented by the offshore team.

This is also a great opportunity to build a good relationship between onshore and offshore, and between business and IT development.

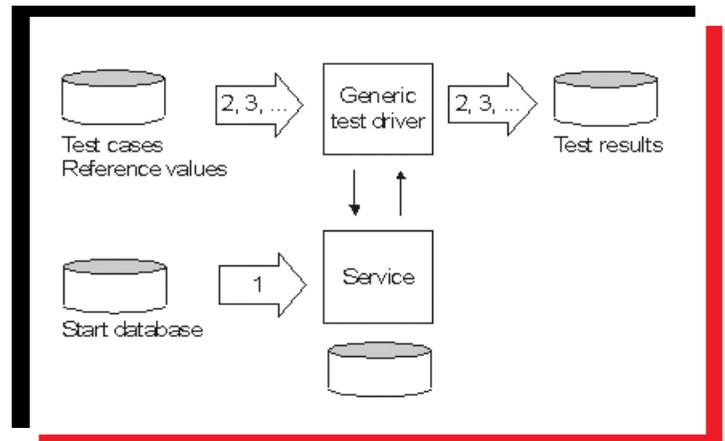


Figure 5: Example: generic test driver for SOA Service (Source: "Enterprise SOA", Prentice Hall 2004)

SOA-Driven Documentation

Finally, we must look at the issue of documentation.

The Agile approach emphasizes face-to-face communication over written documents.

In an offshore situation, regular face-to-face communication is not always achievable.

Consequently, the amount of documents to be written is larger than in the usual Agile project.

THBS has a lot of experience in finding the point of "sufficient" documentation.

THBS puts emphasis on the structure of the documents which are exchanged between the onshore and offshore team.

Leveraging SOA to structure the documentation is very helpful, since the SOA level is sufficiently concrete, without being too close to the implementation.

Using state-of-the-art tools like BusinessGlue's PlanCentral™ to manage the documentation of different SOA artifacts between the onshore and offshore team can be very helpful.

Conclusion

Establishing a long-term trust relationship with our customers is the first important step towards mutual success.

In order to address the 5 key issues in offshore projects that were mentioned at the beginning of this whitepaper (missed cost savings goals, quality problems, cultural differences and communication problems, legal issues, loss of control), THBS has developed an offshore development methodology which combines Agile with SOA.

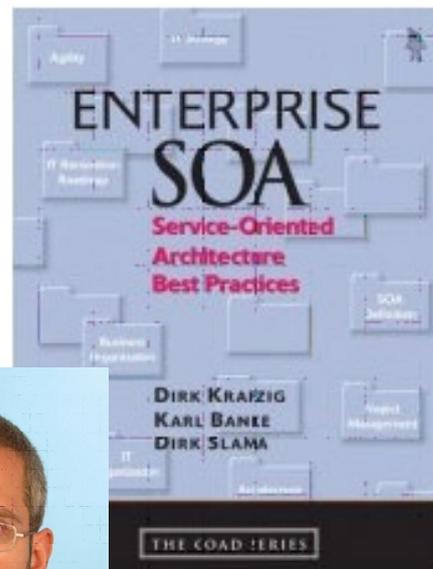
The best practices documented in this whitepaper are specifically tailored to address the risk factors in offshore development by addressing cultural and team issues proactively, by increasing transparency through test-driven development and immediate feedback, and by identifying quality issues immediately through continuous integration.

References

"The Best Practices which we have jointly developed with THBS and which are partly documented in this whitepaper have enabled us to establish a very efficient and risk controlled offshore development process.

This is important to us on two levels: first, because THBS is a strategic partner for us in the development of our PlanCentral™ product. Second, because THBS is also an implementation partner for us in our end-customer projects" says Dirk Slama, Managing Director BusinessGlue GmbH (Germany) and Co-Author of "Enterprise SOA".

"We enable large enterprise customers to manage IT projects more efficiently through our IT Excellence™ IT strategy framework and our SOA-based PlanCentral™ IT planning suite. It was vital for us to find a partner who shares the same principles and methodologies for managing complex IT transition project. THBS has proven to be a partner who understands the mechanisms of Agile and SOA-based IT transformation projects."



Links

[Gar] Gartner Group, Gartner on Outsourcing, www.gartner.com

[Sta05] Standish Group, Chaos Report,

www.standishgroup.com/chaos_resources/index.php

Torry Harris Business Solutions Inc, a US based services provider with a large base of technologists located in the UK, India and China has provided cost effective solutions at a design, development and support level to a variety of enterprise clients across the world since 1998.

The company specializes in integration, distributed computing, and its focus on SOA is a result of nearly a decade of expertise gathered in the middleware space.

The company has partnerships with almost all the leading SOA and integration product vendors.

SOA, involving the creation of autonomous parts of a solution, lends itself admirably to the cost effective model of offshore service collaboration.

A separate white paper entitled "SOA Implementation with an offshore partner" available for download, explores this model in a more detailed manner.

Further information about the company and a variety of white papers on SOA are available at www.thbs.com/soa.

For more information, write to us at soa@thbs.com.

