

# Preparing for Offshoring Initiatives

by Marc Stuessel

Glancing over the brochures of offshore suppliers of software development services, clients often assume that offshoring development tasks will unfailingly result in phenomenal cost savings and software of the highest possible quality. While there certainly is the potential to reap significant benefits, these promotional materials hardly ever mention:

- Additional effort and costs
- Risks
- Practical difficulties in putting offshoring into action
- Difficulty of achieving a quantitative assessment of the actual success of an offshoring initiative

In my experience, the projected objectives are rarely met, and almost never in the first offshore project a client commissions. Successful offshoring takes at least five to six years or at least three to four projects to implement successfully. This is in sharp contrast to the expectations of clients who think that making use of the difference in labor costs alone will save huge amounts of money from day one.

The problem with the idea of a highly competent yet inexpensive supplier is not that this idea is completely mistaken; the problem is

that it is not thought through. Every so often, it turns out that a seemingly simple idea neither is simple to implement nor achieves the expected objectives. In particular, clients underestimate what they have to contribute to a successful offshoring initiative, both in terms of the organizational changes they need to make and the tasks they have to perform.

In this article, I will explore some of the less-well-known aspects of offshoring IT application development:

- Organizational issues
- The project's business case
- Contractual issues

The common denominator among these issues is that initially they cost money and require effort. The amount of money and effort required in most cases goes far beyond what management expects, and naturally these expenses need to be subtracted from the savings the offshoring initiative may realize.

## ORGANIZATIONAL ISSUES

### Supply Management

One of the most decisive consequences of offshoring is that a client establishes a contractual

relationship with a supplier who has fundamentally different profit motives. Clients therefore need to *organize, direct* and *develop* this relationship. The most important activities are:

- **Management of suppliers.** This includes activities such as supplier appraisals and audits.
- **Supplier development.** The client must nurture and support suppliers proactively.
- **Supplier integration.** Client and supplier establish the basis for their interorganizational cooperation by working together closely beyond the scope of individual projects.

The long-term objective is to build a network of highly competent and competitive suppliers. This requires a purchasing department with both knowledge of sourcing strategies and the ability to help projects achieve their technical objectives.

Usually, however, there is no supply management. Instead projects establish and maintain their relationships with suppliers without involving the purchasing department at an early stage. The client classifies suppliers as resources that can be substituted at short notice, and neither client nor supplier takes the long-term perspective that is necessary for a mutually

beneficial business relationship. With no supply management in place, clients squander one of the biggest opportunities for cutting costs.

### Project Governance

Often the executive board arrives at the conclusion that offshoring will result in improved quality or reduced costs — or even both — and mandates that the development of a system take place elsewhere. Yet project managers and technical staff of offshoring projects do not receive management support when it comes to making the project a success.

Establishing project governance is an important prerequisite for successful offshoring. Project governance is the set of structures, systems, and processes surrounding the project that ensure the effective delivery of the project through to full deployment and benefits realization by the organization. Project governance is essentially an approach to risk management. This approach must be in place even before the offshoring initiative starts. Risk management in this context means maintaining the balance between the extremes of risk avoidance and risk abdication.

The organization must have adequate checks and balances to ensure that the project is delivered and meets the organization's risk and compliance regime without an excessive set of process requirements. The implementation of effective governance processes across the whole system lifecycle enables both the project manager

and the organization to deliver the system, reap the benefits, and make the hard choices when required.

Project governance typically requires the following high-level roles:

- The *project manager* is accountable for the management and delivery of the project as specified in the project management plan.
- The *project steering committee* is accountable for the overall delivery of the business system to the organization and reports to the information management group.
- The *information management group* is accountable on behalf of the executive board for the program management and balancing of business systems projects against available resources and the information management strategic plan.
- *IT services* applies suitable project management processes and ensures consistency with IT standards and change control.
- The *business system owner* is accountable for the management of the business systems lifecycle and the realization of benefits identified in the business case.

Implementing project governance goes beyond the creation of a steering committee and selection of a sponsor. It extends across all aspects of the system lifecycle from inception through to replacement. The composition and allocation of

**Implementing project governance goes beyond the creation of a steering committee and selection of a sponsor.**

the aforementioned roles will vary. In a large organization, committees may fulfill the roles. However, in a small organization, the roles may be exercised through the chief executive officer and the project manager. If the questions are asked and checked, then the result is an informal governance process.

The benefits of project governance are that the organization maintains a focus on the business purpose and expectations while still managing the typical technical and organizational change issues that arise during the life of a project. The point is that if an organization with undefined structures and processes offshores, either the supplier's capabilities are not exploited — which results in higher cost and longer schedule — or the client eventually shapes up.

Effective governance ensures that:

- The system lifecycle is managed and benefits are realized.
- The outcomes are aligned with business needs, balanced against the desires of interested parties.
- Risk profiles are managed to acceptable levels.
- Project deliverables meet the planned timelines, cost, and quality.

- The scope and milestones for go/no-go are managed.
- The organization has the capacity and capability to utilize the delivered outcomes.
- Appropriate methodologies are used for planning, design, installation, and implementation.
- The system is installed, stabilized, and standardized, and a process for continuous improvement is implemented before any enhancements are begun.

Clients are often all too happy to hand over the responsibility for a project to a competent supplier. Yet it is the client's responsibility to make sure that they get what they pay for and that the project and the offshoring initiative itself are technical and financial successes.

### THE BUSINESS CASE

How can an organization be sure that an offshoring initiative will yield a cost reduction at all? The tool of

choice for comparing expected results with the actual outcome is a business case. A business case supports decision making and planning, including decisions about whether to "make" or to "buy" in the first place. Business cases are generally designed to answer questions like: What are the likely financial and other business consequences if we take this or that action?

A good *cost model* is one of the essential building blocks of a meaningful IT business case. It is also useful for managing costs and benefits during the course of an offshoring initiative. A good starting point for a business case consequently is to define a complete cost model for the business case analysis.

The cost model is important because the real number of cost items of an offshoring initiative and their specific costs are often unknown to the project staff and management. This is important,

as clients often assume that the promised cost reductions are calculated on the basis of total project costs. However, this is not the case: suppliers advertise their services on the basis of their value proposition, which might make up only a fraction of the entire effort. Let's assume, for the sake of argument, the project cost structure shown in Table 1.

If a supplier promises to cut the software development costs by 50%, then clearly the total cost reduction for the project will be a far cry from 50%. This is simple enough to understand, yet clients get carried away by these advertised cost reductions and, for some reason, assume that the total bill will be reduced by a similar amount.

A cost model enables the project team or higher management to track down all the relevant cost impacts that result from the project while helping to prevent double counting. A good model shows every possible place to look for cost impact, and it clarifies which items and data to omit as well. Models bring together cost or benefit items that have common causes and thus change together (e.g., expenses that have the same cost drivers). This gives management effective "real-time" financial control during implementation and operation of a system.

### CONTRACTUAL ISSUES

For an offshoring initiative to succeed, the system to be developed

Table 1 — Sample project cost structure.

Item	Percentage of Cost
Hardware	3%
Development of software	36%
Installation/rollout	4%
User training	7%
Support	15%
Maintenance	5%
Infrastructure	30%

must be analyzed and specified as precisely as possible before the contractual negotiations begin.

The client's domain experts — who usually specify the system to be developed — have to accept that they cannot commission changes or additional features at will. With process-driven development and an outside implementer, every change and every additional feature will incur costs. Any change to the original specification is treated as out of scope. Usually these additional features are charged at a higher rate than comparable features that were included in the original specification.

In order to avoid these additional costs, a client has to make sure that the domain experts follow a process. This process comprises:

- Specifying features
- Ensuring the quality of deliverables
- Specifying changes or additional features that will be implemented in a future release (“Future release” does not necessarily mean “the next release.”)

While this sounds easy, it is actually very difficult. As every change comes with a price tag, domain experts often worry that changes will not be approved by management even though they are essential to the system. Management has to reassure domain experts that it really will make and pay for these changes somewhere down the line, even if the domain experts forgot or overlooked features in the

specification that formed the basis of the contract. It is particularly important to ease this fear during the phase in which the domain experts are learning how to specify for an outside implementer.

For their part, domain experts must be trained to specify features as completely and in as detailed a manner as possible. Not many people can write good specifications. With in-house software development, this deficiency is offset by software developers who have a grasp of the domain and are able to live with incomplete or even incorrect specifications. With an outside contractor, the client gets what the client specifies. Therefore domain experts will have to learn to live with results that are not perfect until one of the next releases. In terms of cost, it is an absolute disaster when the scope of a specification is changed significantly during the implementation phase.

#### **Legal Know-how**

A good contract is often the key to a successful IT offshoring relationship. Issues such as quality of service, staffing, pricing and payment, warranty and liability, escalation mechanisms, termination, intellectual property matters, and information security need to be formalized. It is important for the client to resolve these issues internally before negotiation starts so that a more proactive stance can be taken during the negotiation. Such a proactive stance can enhance the client's bargaining position and help the client obtain more favorable terms.

**Domain experts will have to learn to live with results that are not perfect until one of the next releases.**

Individual projects that are not embedded in an overall offshoring strategy tend to accept the supplier's standard form contracts. This is usually a mistake, as these contracts are geared toward the supplier's interests. A client has to assume that a supplier has far more knowledge about the legal aspects of an offshoring initiative, and equaling this edge will not come for free.

If the client is not on par with the supplier, they will find that it costs a lot of effort and money to constantly renegotiate contracts over the course of a project. Consequently, a client needs legal advice, at least for the first offshoring project. The legal advisor must have experience with offshoring contracts; it is a fair assumption that a client does not possess the required legal know-how in house and will have to bring it in from outside. Trying to save money by relying on the goodwill of the supplier, their standard contract, or the assumption that they like their clients so much that they forgo their profit margin when the client has difficulties fulfilling their part is a mistake.

#### **Negotiations**

Contracts in most cases take a considerable time to negotiate,

and once implemented, they are difficult to unscramble without incurring a substantial amount of extra cost.

If a well-drafted contract is in place, its effective management should not be too difficult, because the necessary mechanisms should have been built into the contract. The client should assign the contract management responsibility to either the purchasing department or to managers who understand the services provided by the supplier. They will have to deal with routine measurement of the supplier's quality of service against contractual requirements, handle change requests and payments, and assume the overall responsibility of ensuring that the services provided by the supplier are of an acceptable quality. In many cases, the quality of IT systems is critical to a company's success, and hence a direct communication channel should exist between top management and the managers dealing with post-contract management so that problems detected in contract performance are communicated to senior management quickly and efficiently.

### **Specifications**

As I've said, the system to be developed must be specified as accurately as possible before negotiations start. If this is not the case, the supplier will charge more for the additional flexibility incompleteness mandates. Also, every additional feature will be charged at higher prices than the ones in the

original specification. This makes it an absolute necessity for the client to implement requirements engineering well before the offshoring initiative starts.

**Trust and professional behavior cannot be relied upon once a contract is cancelled, because the relationship between client and supplier is cooling very quickly.**

In case requirements are not specified completely at the time the contract is signed, the contract needs to include provisions for the analysis of these incomplete requirements. If this is not planned for, the delivered system will have properties that are determined by assumptions made by the supplier. It is a safe bet that these assumptions will be those that are cost-effective for the supplier.

### **Dispute Resolution and Termination**

Offshoring often involves large sums of money and complicated issues over the contract period. Disputes are therefore not uncommon in the course of a project. Instead of resorting to expensive and time-consuming legal action every time there is a dispute, proper mechanisms should be built into the contract for dispute resolution through an independent third party (e.g., an arbitrator). In cases where the escalation mechanisms

fail, the contract is terminated. One common weakness in offshoring contracts is that the consequences of termination are not taken into account. It is important to address the following issues:

- Buy-back arrangements for hardware and third-party software and a formula for price determination
- Transfer of intellectual property (e.g., via code escrow)
- Transfer of relevant third-party contracts (e.g., maintenance contracts)
- Transfer of data and know-how
- Guarantee by the vendor to assist and cooperate in a smooth handover of whatever work products exist at the time of termination

In addition, there should be explicit agreement about the termination of services in cases where service levels are not met or the supplier goes into liquidation. Trust and professional behavior cannot be relied upon once a contract is cancelled, because the relationship between client and supplier is cooling very quickly.

### **Communication**

For the offshoring initiatives of non-US/UK companies, there is usually a need for professional translation of all kinds of project documentation, such as contracts and specifications. This takes considerable time and money, because a working knowledge of languages other than English is practically nonexis-

tent among offshoring suppliers. This is not a problem as long as the business case and the project plan include cost and time provisions for these factors. Usually they do not.

If the entire system is developed offshore and the targeted market is the national market of the client, then the GUI, system documentation, help systems, tutorials, and operation manuals need to be translated back into whatever the client's language is. Some clients and suppliers use student translators with no domain knowledge whatsoever for this task. The results are usually a disaster and require extensive post-editing.

As vexing as a clumsily translated operations manual can be, there is the potential for a greater problem here. Language skills acquired in school a long time ago are no basis for contractually relevant documents that might result in damages worth hundreds of thousands of dollars. For example, what happens if the supplier implements some requirement according to an incorrect translation? Who pays for the additional effort (or contract penalty) in this case?

This scenario demonstrates the importance of quality-assuring every translation. Offshoring requires investments in the language skills of the staff involved in

a project. Engineers are usually not trained for this kind of work, so if offshoring is a long-term option:

- Authors have to be trained to describe requirements using a standard vocabulary, which must be defined in a company or project dictionary. Ideally a semi-automatic translation becomes possible.
- The client should work closely with a translator who has knowledge of the client's business.

For US and UK companies, there is no need for translations, provided their offshoring suppliers are located in India or are otherwise fluent in English. If such companies decide to offshore to Eastern European countries, however, they might face a translation problem. The knowledge of English is not as good there as in the countries US and UK companies have thus far preferred for offshoring.

## SUMMARY

In this article I have explored some of the less-well-known offshoring issues. Successful offshoring takes time and results in less-than-expected cost reductions because clients need to prepare offshoring initiatives by building up knowledge in various areas unrelated to software development and

establishing structures and processes that — depending on the client — might well be projects in their own right. Clients therefore need to realize that offshoring is a long-term option.

The most important difference between offshoring and inhouse software development is the contractual relationship offshoring brings about. The relationship between client and supplier requires the client to both manage the relationship and deal with contractual affairs that are of no concern with actual employees. Both of these aspects involve effort and costs that are often overlooked when an offshoring initiative is started.

*Marc Stuessel is a freelance consultant with more than 10 years' experience. His main areas of expertise are the design of large-scale Java enterprise systems in the German automotive and airline industries and leading multinational developers implementing these systems.*

*During the last four years, Mr. Stuessel has participated in offshore software development projects with Indian as well as Central and Eastern European suppliers. This experience has given him the opportunity to see firsthand how important the preparation of offshoring initiatives is. Participating in these offshore projects led to his interest in the economic aspects of system development and the alignment of IT and business issues.*

*Mr. Stuessel can be reached at [marc.stuessel@syntesion.de](mailto:marc.stuessel@syntesion.de).*