2005 Technology Journal

Second Edition

October 2005

3. OFFSHORE OUTSOURCING METHODOLOGY FOR SMALL-AND-MID-SIZE COMPANIES

DataArt's SMB methodology for best practices in offshore outsourcing is based on "Business Practices and Lessons Learned" accumulated by DataArt during a successful completion of over 300 software projects and five years of development. At least 90% of these projects were commissioned by small and mid-size businesses in the U.S. and UK. In 2004, based on completed case studies and corporate standards, DataArt has started to formulate a comprehensive methodology that could be shared with clients, partners and subcontractors. We expect that in 2005 will see several projects completed by other vendors adopting the DataArt SMB methodology.

3.1 Roles

To define the SMB methodology for best practices in offshore sourcing, the following roles are instrumental:

- Stakeholders - initiate and define system requirements, serve as a source of specific knowledge for a project team and determine the success/acceptance criteria.
- Account managers - support the project with all the necessary ingredients – personnel, funds, tools and so forth.
- Project coordinator - responsible for communications and quality assurance. This role's goal is to bring together the 'united team' consisting of both client's and developer's representatives. The coordinator's main responsibility is to keep every team member on the same page in terms of system vision, goals, constraints and plans.
- Chief software architect - oversees technology decisions as well as work flow and schedules. Also responsible for system operability after the active development phase is over, including training, documentation and maintenance tasks.
- Developer - responsible for work products based on individual assignments.
- Quality Assurance Managers and Testers - not mandatory in every project. In smaller ones, Project Coordinator assumes the responsibility for functional quality while Chief Software Architect oversees reliability and consistency of the system's components. Normally, Quality Assurance Manager builds a formal quality measurement process while Tester actually runs the process.

3.2 Description

DataArt's SMB Methodology is focused on determining the most effective approaches to small and mid-size projects (up to 10 man-years) and is based on the following principles:

3.2.1 Visibility

All team members should have a clear and consistent vision of business goals, quality and acceptance criteria.

3.2.2 Sustainability

This methodology operates with fixed schedules and budgets. Much emphasis is placed upon the level of prior training for two key team members – Project Coordinator and Chief Software Architect.

There are two types of project-related risks:

- Fundamental, which are political, economic environment, health and which small businesses have little or no control over.
- Operational risks, which impact project budget, schedule overruns and quality issues and will have, on average, much greater impact on a given business. A small business might simply not have time or money to try again. What's important, project managers do have means to manage these risks

Because fundamental risks are hard to manage proactively, this methodology recommends that project managers in small business avoid them, while placing higher attention and more resources on managing risks that are under control - i.e. operational.

3.2.3 Full coverage without redundancy (the Iron Triangle)

All functions, requirements and work products must correspond to the original task without triggering additional expenditures. One of the most popular metaphors in the IT-projects management is the Iron Triangle. It represents three key parameters of the project – functionality (quality), schedule (time constraints), and budget (money). It is impossible to change one parameter without changing the other two. Thus this metaphor implies two types of conflicts:

- Conflict of project's and vendor's interests. It is advantageous for the vendor to follow the client's wishes to add new functionality and to make continuous improvements, but at the same time the vendor might not disclose possible quality decline and increased project risks caused by the project complication.
- Client's desire to get maximum functionality for the minimal budget. In most cases the higher functionality/price ratio leads to higher risks.

3.3 Jump Start

DataArt SMB methodology is aimed at the enhancement of mutual understanding between the client and the vendor and at building a win-win relationship. This idea can be expressed in three ASAP principles.

3.3.1 ASAP (As Soon As Possible)

Amount of time allocated for release of the first functional version of the system should not top 20% of the total allotted schedule and may only be preceded by knowledge

transfer and requirement specification stages. The project development time should be clearly outlined and determined at the start. This allows avoiding some common risks:

- Dismissal, illness or transfer of key stakeholders and developers to another project might lead to the loss of important information. Managing this risk by duplication, detailed documentation and formalization is usually too expensive and ineffective for relatively small projects.
- Shared resources conflicts between client and vendor – for example time on testing servers, potential users, etc.

Immature companies which are aware of these risks often try to protect themselves by overstating the time and budget estimates. This leads to more relaxed work at the start ("we have plenty of time") and doesn't solve the problem in the end: all reserves are spent in the beginning and in the middle of the project.

3.3.2 ASAP (As Specific As Possible)

The most important principle of SMB methodology is maximum relevance of every activity for the real project requirements. Irrelevant requirements become the core of the most devastating risks, which like a malignant growth destroy the solution from the inside and reveal themselves only when it is to late for treatment.

An SMB methodology distinguishes three types of requirements that should be avoided:

- Abstract requirements as a result of different expectations on the part of client and vendor. For example the client wants to develop an important and complicated system using cutting-edge technologies so that the system could remain up-to-date for a longer period of time. Usually such technologies are pretty raw, immature and untried, which leads to the prolongation of the terms and often several remakes of the system or its components. The vendor in turn may decide to try using seven-level architecture, thinking it would work better as it is more complicated. As a result you get the more expensive and longer project, higher risks and all this just for the love of the game.
- Irrelevant requirements tend to appear from the analysis of incomplete and controversial initial requirements. Often the development starts with the creation of engines, libraries, and platforms that are useless on their own, but supposedly help further application development, making it easier and more effective. The first change in the requirements usually shows that the assumptions that were made are false and the libraries created on the basis of these assumptions cannot be used.
- Internal requirements may emerge when a client or a vendor decides to use the existing tools and processes in a new project without a reason. Many companies often try to increase their profits (or reduce costs) using the existing code. This could be very efficient, but in most cases this leads to adding unnecessary functionality, extra dependencies, and inherited problems. Besides it is next to impossible to determine a person responsible for the system's malfunction if it occurs.

3.3.3 ASAP (As Simple As Possible)

From all possible solutions, choose the simplest (the cheapest and fastest to implement) unless there is 100% certainty that this approach doesn't work or doesn't meet the requirements. A typical problem is over complication of the initial system requirements due to the excessively sophisticated architecture. This may seem obvious, but rarely do people check if the solutions really should be as complicated as they sometimes are.

Using the simplest solutions has one more important advantage – developers do not get too "emotionally" involved in architectural and technical solutions they create. If they do, they may end up developing elements of aesthetical and historical value for them, but without actual use for the required system. They might leave these elements in the system, even though it might make the system more complicated, reduce quality and increase risks.

3.4 Single-use Components and Refactoring

If there is part of the system that might often require changes to satisfy yet another set of requirements, it may be reasonable to create a new single-use piece for each situation instead of changing the whole system every time. Although it goes against the popular principle of reusable code, often the module modifications, debugging and adjustments take as much time as its redevelopment.

Such a process is in fact a kind of re-factoring performed not for engineering reasons, but for economical effectiveness. Engineering reasons are rarely clear to the client and even more rarely valued at their true worth, while financial aspects are clear, objective and directly influence the project success.

3.5 Organizational and Communication Principles: Persistent Core Team

Group dynamics and communication are the key factors in managing requirements and projects in general. Effective teams and well-organized communication dramatically reduce the risks.

3.5.1 Team Model

An offshorer working with SMBs must adhere to the single team paradigm which implies that all project stakeholders including the client representatives, executives and involved third parties work together striving to solve their common business problem. All the team members should be aware of the goals and success criteria.

3.5.2 Project Leader

Analyzing our most successful projects, completed ahead of the estimated schedule and with budget savings, we found that each of them had a serious leader. The project leader has the following three characteristics:

- Understanding of the goal and mission of the project, needs and requirements of users, some experience in the application area
- Fundamental understanding of architecture and technology
- High motivation and good temper

Finding, attracting and training leaders is a substantial part of the overall project management process. In fact, there are probably quite a few potential team leaders in your company, but often they disguise themselves by not paying enough attention to their appearance, getting the “wrong” education, not willing to get promoted and so forth.

3.5.3 Group Dynamics

The guarantee of good group dynamics is in the balance of specialization and integration. An SMB methodology implies self-organization of the groups based on particular qualities of each team member.

There is a minimal level of requirements to managers and engineers that exists in the industry. For example, a developer should be proficient with the programming language of choice, be able to estimate the time required for a task, and compare possible technical solutions. Many companies expect their entire staff to enhance all these skills over time. Meanwhile, DataArt found that if the skills of the group are developed asymmetrically, the group dynamics turns out to be more effective.

Roles and responsibilities are distributed according to the natural talents. Thus the organization achieves the necessary level of specialization and at the same time every person feels involved in the project and feels he/she is valuable. Besides, this creates interdependencies between people which helps integration and builds an atmosphere of mutual assistance.

Unfortunately, building asymmetrical groups is rather complicated, especially if the resource pool is limited. It is much easier for a top-manager to have many symmetrical and perfect managers and specialists. In a real situation, there is a sheer shortage of “perfect” personnel and the simplification is carried out by merely ignoring the individual differences. Such practice leads to the drastic growth of risks of divergent trends in the middle and in the end of the project.

3.6 Communication

All communication is conducted with the use of open channels, such as mailing lists and conference calls. The goal is to maintain a sense of real involvement for every team member.

3.6.1 Mailing Lists

There is at least one mailing list for each project, to which all project participants (both on the client’s and vendor’s sides) must be subscribed. For larger and more complicated projects there may be mailing lists created for more specific purposes. The main principle is that it’s better to have redundant subscriptions than to have a person involved in the project not subscribed. All mailing lists correspondence is saved for future references.

People frequently try to avoid using mailing lists, instead communicating via direct e-mails, phone, instant messengers or Skype. This can only be allowed for small and simply organized projects and, even in this case, the key results of the communication should be mailed to the mailing list.

The reason for the high efficiency of mailing lists is that they quickly build the unified culture of communication, and team members help to maintain the integrity of communication.

3.6.2 Direct Communications

All communication should be as direct as possible. This is the fourth ASAP – As Straightforward As Possible. All intermediate links should be eliminated as they can be a source of potential information distortion.

Unnecessary complications may arise when communication practices used before the project launch remain the same for the actual project. For example, an analyst who collects, refines, and formalizes client requests during requirements definition phase often gets so used to this function that she continues to work as a translator even when development starts. This is very efficient, but only until the moment when the first version of the application is shown to the end users. Here the amount of feedback, emotions and number contacts points increases dramatically and this person becomes a bottleneck rather than an amplifier.

3.7 Summary

The uniqueness of this methodology is that it solves a business task: to achieve a desired level of quality with controllable risks, while keeping limited schedules and budgets.

3.7.1 Pros/Cons

Pros:

1. Helps achieve project goals amid rigid constraints.
2. Provides maximum business performance for invested funds.
3. Places high requirements only on management staff (coordinator and architect)
4. Provides for maximum process flexibility using client's resources in the most efficient manner.

Cons:

1. Doesn't work for large projects where it is more effective to manage the risks, not to avoid them.
2. There is virtually no hope to "get a perfect product the first time around".
3. Demands constant involvement from stakeholders (this is typical for all agile methodologies) or very experienced vendor.
4. Demands understanding of a complete business model on the part of a developer.
5. This, in turn, requires trust on the part of a client and substantial efforts for knowledge transfer.

3.7.2 Implications

This is business engineering rather than craft or scientific approach - bringing technology closer to business. To this date, science and art have produced few pure technologies. Creativity is as important as ever. How come a student of math can absorb in just a couple of years the knowledge that took centuries for world's best minds to invent? The methodology of science has progressed greatly and allows us to represent complex concepts in short and elegant formulas. The same is taking place in computer science – what took large scientific team years to complete can now be done by one programmer. Nearly everywhere - except for the cutting edge of technology business - there is no point in 'reinventing the wheel'. We're better off using the most adequate instruments from those already available. Therefore, more and more problems can and should be solved with minimal investment and maximum effectiveness.

Author: Eugene Goland, President

Eugene Goland is an expert in offshore outsourcing for small and medium businesses (SMB). He is a founder of two profitable technology companies with offshore operations. Goland has a strong hands-on knowledge of the outsourcing industry, from both supply and demand sides, and has been advising SMBs on the issues of outsourcing. He holds an Executive MBA from R.H. Smith.

eugene@dataart.com

212-378-4108 x. 17