

SHORING UP THE OFF-SHORE PROJECT

A View from the Trenches

There has been a great deal of press lately about the use of off-shore resources for IT development projects. Based upon experiences that I encountered on two recent projects for different clients, there are both benefits and challenges for project managers involved in managing such projects. While my experiences should not be taken to reflect upon all off-shore teams, stories shared with fellow PMs led me to believe that my experiences were not unique. As such, I hope that sharing my observations and "lessons learned" will help other PMs who are embarking on projects using similar resources.

Overall, I found the off-shore staffs to be friendly, intelligent, enthusiastic, and more than willing to put in extra effort when schedules got tight. Both teams were comprised primarily of young, well-educated men, and the personnel costs were low, the key driver for the off-shore trend. English was generally well understood and spoken, although thick accents occasionally led to difficulty in verbal communications. Both teams had good programming skills and, because of the time zone differences, were often able to make code changes "overnight" and have updates available for the U.S. based teams the next morning. However, distance, differences in training and experience, and communications provided challenges in ensuring that these resources were effectively utilized.

Lesson 1: If there is a lack of a commonly understood development process, the team leads should identify and agree upon their deliverables early in the project life cycle.

Although both of the off-shore teams with which I was involved claimed to have mature processes, the actual practices fell short. This led to initial confusion within and between the teams as to who was responsible for what. It also led to misunderstandings about what was to be included in each project deliverable. While this problem is not unique, it is compounded when part of the team is a dozen time zones away.

When defining the deliverables, include the completion and acceptance criteria for each. Enforce these criteria – especially early in the project when the teams are learning to work together.

Lesson 2: The use of video conferencing tools and shared work spaces, such as electronic whiteboards and co-authoring tools, helps to supplement limited communications.

Weaknesses in processes or project deliverables can sometimes be reduced when open and frequent communication exists between team members. This is difficult when dealing with remote teams. On my projects, the communications tended to be formal conference calls or informal e-mail messages. On the conference calls, many participants were half asleep because they were a dozen time zones away and had already put in a full day of work. The e-mails were often message threads with a day's delay between responses because of time zone defined work schedules. In neither case was the communication optimal. Once introduced, however, the use of the supplemental communications tools helped make the experiences more interactive and productive.

As a side note, many foreign workers and offices have limited access to high-speed telecommunications. Even if the necessary services are offered in foreign countries, acquiring them can often take months.

Lesson 3: Build shared experience at key points in the project. This is preferably done in person.

Getting the team leads together for the initial project meetings was critical. The initial shared understanding of the project's goals and domain level requirements was more than worth the expense. Team meetings at the end of each development iteration were also extremely beneficial. My meetings included team leads, but not the full team. In retrospect, I would have pushed for a broader audience, even if some had to attend via video or teleconference.

In addition to kick-off and review meetings, I would especially recommend having architects attend the project domain level modeling sessions in person. Off-shore development teams are fine, but architects need to have extensive face-to-face time with analysts and project/problem domain experts. This can not be easily replaced even using supplemental communication tools.

Lesson 4: Put an extra emphasis on continuous architectural review and enhancement.

On both of my projects the team's programming skills were good, but there were few skilled, senior-level architects. After the initial product design, the original architects

often moved on to other projects and left future architecture decisions to less experienced designers on the team. The development teams tended to throw programming resources at problems before considering design impacts. The resulting code had the potential for being poorly designed, brittle, and difficult to maintain. I found that I had to be constantly vigilant against the short-sighted "quickie" solution.

Lesson 5: Review your assumptions about "common understandings."

An issue that arose with the foreign teams was the lack of "shared experience" with their U.S. counterparts. This was much broader than just project/problem domain experience or even general U.S. business process experience. Items which I've rarely had to review with U.S. coders had to be addressed in detail with foreign programmers including basic Windows design standards. This "shared experience" gap required much clearer and more detailed documentation for some items which I generally assume to be understood by U.S. coders.

Lesson 6: Stay on track by implementing and enforcing an iterative development life cycle. Time-box the efforts to control schedule creep, and keep the iteration durations short.

As with any group of people, I found that subtle differences in understandings of requirements led to different visions of the objective. The geographic separation of my teams increased the difficulty in discovering these differences until they were reflected in code. Introducing an iterative life cycle helped to ensure that products didn't get too far off target before problems were identified and corrected. This would have been especially beneficial at the beginning of the projects when the teams were just learning how to work together.

Lesson 7: Build a schedule including national and regional holidays.

Not only were foreign national holidays different than those in the U.S., but there were regional holidays as well. Some of these lasted weeks and had the potential for severely impacting staff and support resources during those times.

The challenges that I encountered were not unique to foreign development teams, but were exacerbated by time/geographic distance and differences in culture and experience. Generally, these challenges had a basis in communication problems and weak inter- and intra- team processes. The earlier these are considered and addressed in your project, the better the chances for a successful collaboration between local and remote teams.

Craig D. Wilson