Strategies for Software Outsourcing Success

Audience:	Business decision makers evaluating the benefits of outsourcing software development and who are in the process of selecting a partner
Author:	John Douglas - john.douglas@mycosservices.com
Date:	Feb 4, 2008

Overview

The outsourcing decision, or the "the pill" as one friend described it is a challenge. When you are looking at someone across a desk, you have a fairly good sense as to whether they are going to contribute to the bottom-line or subtract from it. Introduce the challenges of language, culture, time zones, currencies, and project management and that assessment is decidedly more difficult. I offer here a few thoughts on the decision making process when selecting an outsourcing team, and, more generally, when considering whether to outsource your software development. These observations are based on my experience as a software engineer from the U.S. owning and running a software outsourcing company in Thailand for 6 years. This is all informal, from the hip, prescriptive without explanation, and self-promoting, but also direct, honest, and hopefully useful.

Are You Ready?

You can save money, augment your team, hand-off undesirable tasks, and reduce your organizations overhead by outsourcing. It works. It is growing. There are major successes, and there are major failures. The foremost question, on my view, is the following: is your organization on board? There has to be buy-in within your organization. We've seen projects misfire when on-shore organizations did not have that buy-in from there on-shore team members, ranging from developers to CEO's. There are a whole range of psychological/emotional issues surrounding outsourcing that can limit the success of the outsourcing endeavor. It's fairly easy to boat-anchor a project that you have influence on, so carefully consider your organization's willingness to outsource.

Do the Outsourcing Math, Carefully

The cost savings of outsourcing are not: savings = onshore resource wages - offshore resource wages. Factor in on-shore management time, putting requirements in writing, and logging issues that you might previously have demonstrated to an on-shore resource in an office. These management tasks increase the on-shore cost significantly and need to be weighed, or at least estimated, in order to effectively assess the overall cost-benefit. Consider carefully who will be the first-level interface with your offshore partner. In most circumstances, it makes little economic sense to have a CTO earning $150 per hour directly manage on offshore resource making $12.

Go with an Outsourcing Team Not Just a Country

Geography, language and time-zones are important considerations when selecting your team. However, limiting your selection based on a country is something like building a basketball or football team based on nationality - saying you only want Brazilians or Brits. There are strong software outsourcing teams in India, and terrible ones too. Nothing about being ethnically Indian, on my view, makes you an innately good software professional. The same can be said for any ethnic group or nationality. As an outsourcing vendor, whenever I hear, "we're looking at country x," I try to emphasize the unusual nature of software engineering, where one team can be orders of magnitude more productive than another, and stress that locating *that* team whether it be in Bangalore or the outskirts of Ulan Batar, should be the primary decision making criterion.

Find a Specialist as an Offshore Software Vendor

For any vendor, it is tempting to say "yes" to every question regarding skills/competency. Sadly, many do. At Mycos, we've specialized in Microsoft technologies for 6 years, since before .NET was released, and I know we have some top-notch engineering talent. We are in the process of expanding to Ruby on Rails, but frankly, just maintaining professional proficiency on the .NET platform is challenging, given its rapid development. Ask for resumes of resources available to work on your project and verify that the experience is sufficiently focused in the areas required for your project and test accordingly, even if it is via email or a chat application. With a compiler, the internet for reference, and an infinite amount of time, any resource can complete your project. Finding a specialist, however, is a better approach.

Measure and Track Work

Most offshore software development teams don't have the same fear of micro-management you might find in an on-shore team. Culturally, accounting for work is more acceptable and the general principle of "hire good people and get out of the way" is less effective. Make it clear from the outset that accountability is a built-in part of the project, and is non-negotiable. I'm frequently asked about *how* to track projects and tasks. There are lots of tools: TFS (a Microsoft product), BaseCamp, Unfuddle, Bugzilla, Rational tools, etc. IT professional wage long discussions about the relative merits of the tools they use, but, on my view, it's an insignificant difference. Pick one and use it. If the project is small, have the resource(s) send a daily update via email until you settle on a tool. Account for tasks and hours. The most common failure for offshore software projects is not the limitations of the communications mechanism, but the failure of the stakeholders to use it.

Provide Opportunities for Exceptional Performance

From tracking and measuring, it does not follow that your offshore resources must be regulated like brick-layers. Provide enough space or higher level task definitions to allow exceptional resources to do exceptional things. The arguments, or better consensus, that *we* (onshore, westerners) innovate and architect and *they* (offshore resources) lay the bricks is a self-fulfilling prophecy, on my view. Innovators and exceptional performers are everywhere, and providing them opportunities to demonstrate those skills within the context of your project benefits everyone.

Assess the Relationship Early

Review progress and provide feedback early. Often the small flaws in the process early on become chasms down the road. It does take time time to install tools, become familiar with vertical and domain specific code, and review existing applications and code, but the first bits typically need to be back in front of your eyes in less than 10 working days. I really believe anything longer jeopardizes the project. As soon as you have something to review, promptly provide detailed feedback. This is the most valuable opportunity you will have to make your expectations clear. Software outsourcing is not unlike any other type of business relationship. Setting clear expectations early provides the best opportunity to build a long-term partnership.

Build a Relationship with Your Software Vendor

Especially if you expect the project to be on-going or might have other requirements for the vendor, make an effort to build a relationship with your outsourcing partner. Learn the name(s) of the resources, find out their likes and dislikes. Visit on site if possible. As soon as it becomes human-to-human interaction you immediately take precedence over other clients. Critical deadlines are no longer abstract, urgency is human urgency, and escalation is a phone call away.

Conclusion: Avoid the Commodity View of Software Outsourcing

Software engineering is still not a commodity, on my view. The complexity of the deliverables and the extreme variation in engineering skill levels make it extremely difficult to grade the service, in a manner such as that by which diamonds are graded. Humans interaction adds to the complexity - the psychology of management dictates that resources want to be appreciated, not only compensated. Take the time to assess your potential vendors, find the strongest team, check resumes and references, and treat offshore resources as people. These steps will maximize your organization's ability to be successful in outsourcing your software projects.