



Reaping the benefits of global development

Jason Cole

August 07, 2006 (Computerworld) Offshoring, nearshoring, backshoring — if your company is trying its hand at global development — and if it isn't, then someone is going to ask you about it soon — then you may find yourself wandering the halls asking, "Has anyone seen my developers? Oh, and while you're looking, do you think you could try to find those cost savings and productivity improvements that I was promised?" Global development holds great promise — of reduced labor costs, round-the-clock business hours and greater productivity — but it brings unique challenges that many organizations fail to accommodate in their development plans. So whether you're just dipping your toe into international waters or you already have a development team that spans the globe, I have a few suggestions that might help you get a bit more sleep.

Assuming that you've already found your development partner and have the basic infrastructure in place, the success of your global development effort will rest upon three pillars: communication, team structure and mindful portfolio management.

Communication is the most important dynamic in a global development effort. Forget about the language barrier, though — that's the least of your worries. Your real concern should be the direction, form and timing of your communications, not their language. If you're in Boston, it doesn't matter whether your development team is in Dublin, San Jose or Bangalore, India. As long as the developers are in a different time zone, then the majority of your communications will be one-way and asynchronous. And no matter what that video-teleconferencing salesman may have told you, this means that you and your team will need to learn how to share complex ideas with people you rarely see or even talk to. This leads to a heavy dependence upon e-mail and the written word in general as the primary mode of communication. Unfortunately for you, the fact that you e-mail the guy two cubes down to ask him if he's ready for lunch doesn't mean that you or your team are ready to express complex technical concepts in writing.

Remote communication also introduces the dreaded "decision lag." Time-zone differences increase the turnaround for even the simplest decision from a matter of minutes to a day or more, and heaven help you if someone has a follow-up question. We don't realize how much face-to-face conversation, debate and problem-solving goes into a software development project until we can't have it. This dramatic shift has an immediate impact on technical specifications, troubleshooting techniques and your ability to handle requirements changes, and it needs to be consciously managed.

In the global development model, requirements must be precise, detailed and unambiguous. The business drivers behind the requirements must also be clearly explained, but at a level that can be understood by a stranger to your business. After all, intelligent as they may be, most of your remote developers will have little exposure to your business, no contact with your business sponsors and no presence at any of the meetings where these decisions were made. It's unfair, — not to mention unwise, — to expect them to understand your business well enough to make critical design decisions. If you're lucky, ambiguous requirements will bounce back to you faster than a motorized yo-yo, accompanied by a slew of questions that will all need to be answered before any progress is made. If you're not, then the remote team members will just make their best guess and you won't find out about it until they deliver the application to you with crossed fingers.

To account for this increased need for specific requirements, you need to beef up your local analyst resources at the same time that you add to your remote development ranks. In an average development organization, every two to three development jobs shifted offshore will require one senior technical analyst in the home office to gather and document business requirements and convert them to detailed technical specifications. Over time, a truly effective organization may be able to increase that ratio to 5:1 or greater. These analysts are your interpreters, translating the fuzzy jargon and unspoken needs of your business into a clean technical blueprint that anyone can follow. They are your first line of defense against long-distance chaos.

The second line of defense is your team structure itself. Many companies fall into the trap of thinking that they can run their remote development like a body shop, keeping the senior technical talent at headquarters and farming out specific pieces of work to a cloud of junior developers scattered across the world. Others simply hand off a technical specification to an outsourcing partner and sit back to wait for the results. This assembly-line approach to development might work in specific instances, but most of the time it's the stuff of outsourcing horror tales: *The Project That Wouldn't Die* showing as a double-feature with *The Budget That Ate Chicago*.

The saner approach is to manage each of your distributed teams as semiautonomous development nodes, each with its own senior technical leaders and mentoring structures, but guided by a central technical authority located in the home office. This has several benefits:

- It reduces the pressure on your senior technical staff at home, freeing them from late-night conference calls and the fear of finding their in-box stuffed with questions every morning. It also allows them to concentrate on your most challenging and strategic projects (more on this in a moment).
- It significantly cuts down on decision lag, since local leads can answer pure technical questions immediately.
- It increases retention, a growing problem with offshore development companies, because junior developers have both a mentor and a living example of their potential career path sitting right next to them.
- It increases the quality of the final deliverable, as team leads and architects build and enforce a common technical vision across the distributed network of developers.

The ideal team structure, then, consists of a central development team in the home office — with senior technical talent, strong project management and a team of articulate technical analysts — linked to a network of remote development teams, each with their own local technical lead overseeing a team of junior developers. While this may sound more expensive than the body shop model, it significantly reduces your risk of seeing projects spin out of control and take your career with them.

Even when your global development organization is firing on all cylinders, you still have the potential to fail spectacularly if you don't mindfully manage your project portfolio. Some projects lend themselves to global development efforts, while others cry out to be kept at home. You need to know the difference, or else that high-profile new product launch could be your swan song. As you look at your list of upcoming projects, ask yourself these questions about each one:

- Is this brand-new development or an enhancement to an existing application?
- Is the existing code disorganized and mysterious, or clean and clearly documented?
- Will we be working with unfamiliar or cutting-edge technology?
- Are we building this for a new customer or business unit?
- Are we going after a new market or customer with this product?
- Does the project have a hard deadline or other time pressures?
- Is the project critical to the company's strategy, or even something that will give us a significant competitive advantage when it's complete?

Generally, your remote development teams should get the simple, well-understood projects with the nice, long timelines (of course, if you ever get one of those, they may have to fight your local team for it). In lieu of that, assign maintenance releases, production support and architecture upgrades to the remote teams. They are ideally suited for distributed development efforts, allowing you to reap the benefits of the global model.

The more critical, sensitive or hurried a project is, the closer you need to keep it to the source of the decisions and decision-makers that will drive it. Global development teams simply can't provide the flexibility and agility that you get with a single team working in one place, with one purpose. These projects need the immediate feedback of face-to-face conversation, the power of brainstorming sessions and intensive workshops, and the ability to integrate last-minute changes driven by a dynamic market or shifting competitive landscape. These projects are also more likely to have the attention of the senior executives in your organization, and you can never underestimate the importance of allowing them to see people hard at work on the things that matter to them.

The move to a global development model is a cultural shift, but it is the corporate culture that must change, not society at large. If your development organization can absorb these changes and become mindful of the way you communicate, how you structure your teams and how you segregate your projects, then you will be well on the way to reaping the benefits of successful global development.

Jason Cole is the IT Governance Practice director for Ascend Consulting, a Boston-based IT management consulting firm. Contact him at jcole@ascendconsult.com.