

Management & Teams

Have you ever wondered why working on a distributed development project is so difficult? I have pondered that question as I've participated on multi-site and virtual development teams spread across several time zones, and I would like to share what I've learned about how to make coordination and collaboration of teams easier. In the process, I'll describe some of the practical ideas I've found effective and relatively easy to implement.

New and emerging shifts in how and where we work are allowing many organizations to experiment with multi-site development: the rise of virtual collaboration tools, the use of the Web and the Internet to connect development locations, and the formation of virtual teams. Many of the organizations that utilize these methods and technologies are now ex-

Distributed Teams

One method for bringing your multi-site work group together *by John Suzuki*

▶▶ QUICK LOOK

- Common barriers to coordinating distributed teams
- 6 guidelines for team-building sessions

periencing the coordination difficulties that often accompany distributed development efforts.

Coordination, as we'll use the term here, is simply the managing of dependencies between activities. In a software development environment, coordination depends on two things. First, there must be a common definition or goal, and a common understanding and direction of the project. Second, the software development process requires that all team members strive to build and organize the various parts of the system they are working on so they fit and function properly. Various supporting functional groups within the development team—Testing, Configuration Management, Hardware Engineering, and Program Management—must share information and coordinate design and development activities as the project proceeds. As team members work together to integrate the information and activities needed to hand off the application to the customer expeditiously, coordination is at the heart of the process.

Multiple Sites, Multiple Challenges

While multi-site work groups have an array of tools and technologies to help them improve coordination and productivity, they face some common barriers that can be addressed only through perceptive and inventive management. Here I'll share some of the obstacles I've run into while

working on multi-site development teams.

Loss of Ad Hoc Communication Most teams rely to some extent on formal, organized meetings and status reports. But many of the teams I've worked with in the past have used unplanned, unscheduled contact and *informal* communication as one of the primary mechanisms for transferring project information. These casual conversations in the break room, at lunch, or around the water cooler help team members understand how they do their work, what their most important issues are, location-specific terms and language, and the various responsibilities and expertise of other team members.

Multi-site work groups introduce distance and separation into the communication mix, setting up a range of potential roadblocks in the exchange of ideas between sender and receiver—and affecting both formal and in-

I no longer had informal access to these key individuals, and couldn't rely on them as personal aids for finding more bugs.

Emergence of Unique Subteam Perspectives Physical separation seems to foster a unique subteam perspective. I see this manifested in a resistance to use the other locations' set of internal practices even though it will save time. I personally experienced this once with a team at a distant location that refused to use a standardized development plan template. Instead, they created a new template from scratch and used it to create their own plan. But it turned out that their plan template was missing some key document sections and information required by the other teams on the project. This unwillingness to share information and knowledge—a pattern some folks call the Not Invented Here (NIH) syndrome—is often caused by a lack of trust and a lack of


technology can sometimes be intrusive and disruptive. For distributed teams, the telephone's real-time advantages can be mitigated by differences in time zones. The time zone difference between your New York and California offices, for example, effectively reduces the window of communication to less than five hours a day. Teams split between Toronto and New Delhi would have an even smaller window.

Once, on a multi-site team, I had a team member who rarely picked up his phone messages; his communication preferences centered on email, and that was the only way team members could communicate with him. Email has the advantages of speed and documentation, but it brings with it its own challenges. While generally less intrusive than phone calls, email is often not read or answered in a timely fashion—due in part, perhaps, to the difficulty in gauging the urgency and importance of each day's load of incoming messages.

Some team members feel most comfortable and effective talking face-to-face, as they would in co-located teams, and that's also the method most organizations prefer when communicating important messages or information about the project (e.g., layoffs, site closures, reorganizations, or project cancellations). Because it relies on travel and other logistics, this communication method lacks the immediacy of telephone or email—and in many multi-site organizations it's reserved only for Upper Management staff who have the authority and budget to travel to the various remote locations.

When there are differences in communication preferences, team members may be reluctant to communicate with each other. While these differences usually get ironed out quickly in a team that is in one location, separated teams have fewer opportunities to work out the preference difference—which leads to communication delays and a higher cost in coordination.

Personal Work Style Differences Just as communication styles can vary widely between team members, the effects of personal work style differences can be magnified in a distrib-



The effects of personal work style differences can be magnified in a distributed team setting.

formal communication. Management can usually deal with the loss of formal communication through regular project reviews and offsite status meetings with the different locations—but the loss of informal communication is more difficult to deal with and quantify from a managerial perspective.

It's important not to underestimate the impact of this loss. I have often relied on informal hallway chats with developers who had the reputation of writing the least buggy code, or who knew the most about a particular complex portion of an application. They often gave me insight into the cause of the bugs that we found during the various test cycles. On one particular multi-site team, I didn't have the opportunity to meet many of the developers located at another site.

familiarity with other team members. In my experience, this behavior is often unintentional and arises naturally from separation and cultural isolation.

Diverse Communication Preferences Different people have different methods of communication that they find comfortable and effective. It's important to understand these preferences in the early stages of a distributed project, because the physical separation only exacerbates the effect of these natural differences.

The telephone is a popular tool for quickly resolving questions and can convey more nonverbal meanings than written communication alone. Its effectiveness, however, depends on the other person's availability to pick up the call when you need them, and the immediacy of the

uted team setting. Some developers prefer to work collaboratively, while others prefer to quietly work alone. Other team members prefer the analysis and review of documentation over making appointments to question and talk to other team members, especially if they reside at another location. People who prefer collaborative approaches will have less opportunity to do so in this environment, as the nature of multi-site development generally reinforces the pattern of developers working independently on their assignments and coming together infrequently to report their experiences and progress. The development manager also has less visibility into what each engineer is doing at the remote locations, making it difficult to identify and counteract the barriers that impede coordination.

Inability to Recognize Communication Gaps Technical folks are notoriously unaware that effective communication

is missing, that communication needs to occur more often, or that the current forms of communication are not as effective as they appear. Sometimes it's hard to realize that email, teamware, and other collaboration tools are not enough to facilitate effective teams, processes, and organizations. At times we need to step away from the logical and technical side of development and acknowledge the difficulties with communication from a human perspective. Why is it hard for us to see these gaps? It may be partially a result of focusing on technical training and not realizing the importance of the human side of software development. Software development, says author Gerald Weinberg in his book *The Psychology of Computer Programming*, continues to be a human performance and social activity. Proper communication plays a large role in the relationships of individuals and teams—and ultimately in the success of software projects. Many technical problems, explains Wein-

berg, are actually human problems in disguise.

Finding Solutions

In my experience with distributed teams, we've addressed many of these challenges through traditional techniques that recognize the special problems of multi-site work groups and promote better coordination and communication.

- **Formal communication methods**, such as weekly development team meetings to discuss issues and report on status, are not tools you have to abandon just because your people are spread across three time zones. These are most effective when video conferencing is used.
- A **private videoconference** between the development manager and the teams at the other locations is an extra step that can help identify and resolve project issues. It has the added benefit of alleviating the feelings of isolation that are unique to the remote teams.

PERSPECTIVE

Programming Perestroika

Maps can be deceiving, says Andy Streich; sometimes distance has nothing to do with geography. A veteran of several intercontinental work groups, Streich has seen teams located an hour away from each other completely fall apart, and has watched groups on completely opposite sides of the planet come together in amazingly effective partnerships.

"In one of my early experiences," he recalls, "I took over a project that had inherited a code base co-developed by two separate teams." Separate, it turns out, was an understatement. Pretty soon, says Streich, it became obvious the two halves of the development group had been working with competing interpretations of interfaces and their functions, problems aggravated by different time zones and different debugging environments. "The whole project," he says, "was unpleasant for everyone concerned. There was a lot of blaming when they tried to bring the two halves together, and the product naturally wasn't as good as it could have been."

But he's been able to turn some of the very things that *didn't* work in that experience into assets for other projects, including work with several successful US/Russian teams. Disparate time zones now proved to be an advantage. "It was a positive component for those teams," Streich says, "allowing us two shifts per day. And that can help you achieve great turnaround on defect corrections." If, that is, you've divided your teams and tasks wise-

ly. He advises that work that's very modular, with low coupling between work assignments (e.g., suites of tests), is the best candidate for far-flung teams.

Streich split his development and testing functions; while his US developers slept, his Russian testers would be testing the day's code, and would have it back to California in time for breakfast. If you instead go the route of splitting your programmers, co-developing modules in separate locations, then integrate often and early. He also cautions that you'll want to be very clear about what features of the development and test environments must be identical to avoid confusion.

And while regularly bringing all the distant team members together would have been an ideal tool, Streich has found a variation on that theme to work very well: rotating several staff through the two team locations. "We rotated Russian staff over to California for three-month periods," he says, "so we had someone on our site all the time who knew everyone in Russia. They picked up part of our culture, relationships were strengthened, and the product was better for it."

Even when real face time isn't an option, says Streich, train yourself to operate in that person-to-person mindset—assuming the best intent with every email, and using the phone whenever possible to handle potential conflicts. "You'll save time, build more goodwill, and," he offers, "ship a better product." —A.W.

■ **Workload balancing** and **task alignment** are key. If it's practical, assign development team members to projects based on their individual work preferences and work strengths as another means to enhance coordination. Failing to balance the workload in a distributed development environment may result in one remote location waiting for deliverables from another remote location—which can take a toll on project costs and production schedules.

These methods take advantage of communications technology and tried-and-true management techniques in an effort to bridge the efficiency and morale gaps intrinsic to remotely located team efforts. But I have found that the most important and effective technique for rising above most—if not all—of the coordination barriers we've discussed doesn't require high-speed computer connections and high-tech solutions. The most successful coordination technique, it turns out, is bringing team members together occasionally for face-to-face communication.

Making Face Time Work

"But wait," you might argue, "I thought one of the reasons to have teams working in different locations was to *cut down* on cost." A fair point...but consider the costs that are less evident on a monthly balance sheet. When we first organized one multi-site team, for example, we made the mistake of concentrating on the wrong costs, and we waited six months after assembling the team before we brought the team members together for a team-building session. In retrospect, that decision proved to be costly in terms of productivity. How much we would have gained from moving that meeting up on the calendar—in terms of efficiency, morale, and the payoffs from better communication—is something we can only speculate about. But my experience has shown me that the benefits of having a face-to-face team-building session early in the project cycle can be dramatic.

These sessions can sometimes repair problems you don't even realize are there. A few years ago I was brought in to work with a multi-site



My experience has shown me that the benefits of having a face-to-face team-building session early in the project cycle can be dramatic.

development team. Communicating via email and regularly scheduled videoconferences appeared to be reasonably effective for this team, with no noticeable difficulties in that part of the process. But after the team met face-to-face for the first time for a team-building session, the relationships between the various team members and the different locations changed. Team members noticed that everyone was working better together. There was an improved sensitivity and understanding of the problems that each location had to deal with, and a new respect and acknowledgement of the competency of the members at the other sites. Project problems were resolved significantly faster after that team-building session, and we attributed the improvement to an increased familiarity with each other and increased trust among the various team members.

Although there are many formats for structuring effective sessions, I have found these six guidelines to be the most effective.

1. Hold the team-building meeting offsite, away from the work environment. The most effective team building occurs when phone calls and impromptu visitors are not constantly interrupting team members. Being in a room down the hall from their regular offices means there is always the tendency for your session participants to check voice messages and email, interrupting the session's flow. Although it's sometimes difficult to reserve a large block of time for the entire group, the payoff is usually well worth the effort.

2. Conduct actual working sessions to resolve key team issues and develop common processes. To make the team-building session more effective, I recommend scheduling specific development team tasks that will actually be worked on by team members at the offsite meeting and allow them time to discuss common team practices and processes before returning to their respective locations. This lets the team work together on a small but real project in a safe environment. It also promotes a common understanding of the project values and goals and helps gain buy-in from all team members. In one of our working sessions, we were able to get various team members from different locations to agree on a format for a document template in less than one hour. The same process performed remotely by email normally takes several weeks before approval occurs.

3. Have a trained facilitator assist with team-building exercises. Sometimes additional insight into your team members' work styles and communication differences can be offered by specific team-building exercises. After our team members had worked in various sessions together, we brought in a professional moderator to facilitate an activity that helped highlight our individual perspectives and needs. That sort of understanding can help offset the communication and coordination risks that affect distributed teams; in our case it led to a noticeable improvement. There are several profile tools in common use (e.g., The Birkman Method, the Personal Profile System,

and the Meyers-Briggs Type Indicator), and many human resource personnel or organizational development consulting groups have skilled personnel that can administer such profiles.

4. Plan for extended meetings. I noticed that it took several days before the barriers began to open and direct communication began to flow during

understanding of the expectations of the management chain.

These six elements will help you pave the way for better communication on your multi-site team. For example, the meetings help reduce the cultural barriers between the various locations. The different sites are often entrenched in their own way of doing development based on the local

scribed here can be implemented in any group or organization with minimal cost, and they help address one of the most valuable assets your team will draw on—the relationships between people. Although tools and technology do help to improve productivity within a team, the smartest game plans for improving coordination relate to the human aspects of working together. And the most important method for improving coordination between groups separated by distance is to sometimes physically bring the parties together. Well-timed face-to-face meetings prove to be valuable in promoting group norms, increasing sensitivity among team members, and improving communication. I feel strongly that any expense incurred by the development organization in bringing the team members together from separate, even distant, locations is far outweighed by the improved coordination that a face-to-face meeting makes possible.

From a managerial perspective, better coordination makes it easier to manage and lead the various teams at different locations. These improvements translate into the productivity gains and reduction in development costs that are important factors to organizational success. *STQE*

John Suzuki, president of JKS & Associates (johnsuzuki@jksassociates.com), provides software consulting services to a wide range of industrial sectors in software process improvement, software quality assurance, project management, and verification and validation. He has additional expertise in developing and validating critical software for the medical and pharmaceutical industries.



Improved coordination reduces costs and creates a more simplified and uniform approach to performing development across the organization.

our offsite meetings. This seems reasonable, since trust and familiarity usually take several days to develop among strangers. Be sure to allocate enough time for the team meetings to allow trust to develop naturally—day-long meetings over several days.

5. Take advantage of natural opportunities for connecting. Structuring our sessions so team members could share lunch and dinner really accelerated the connecting process. Those occasions' more informal and relaxed atmosphere fostered a more honest communication among the team members. It was also a good way to learn more about the background and cultures of various team members.

6. Include Upper Management participation. A great way for team members to meet Management and other organizational representatives is to invite them to a separate working session. The working session can often include separate meetings with several levels of management and other organizational representatives with whom the development teams will interface. These sessions often prove very valuable for gaining a common

norms. This is not conducive to fostering or promoting coordination. Also, the meetings seem to improve trust and communication. Part of the improvement comes from an understanding of the communication and work style differences of the team members, as well as from an improved sensitivity to the challenges that the many locations face on each project.

Conclusion

There are several short-term and long-term benefits to your organization's projects from an increased emphasis on coordination. Better communication among the various locations and among the various team members leads to fewer surprises as development progresses. Improved coordination reduces costs and creates a more simplified and uniform approach to performing development across the organization.

In this article we've looked at some common barriers to coordination, and practical methods to improve coordination in multi-site development organizations. Many of the techniques or processes that I de-

STQE magazine is produced by *STQE* Publishing, a division of Software Quality Engineering.