

# Rothman Consulting Group, Inc.

- home
- services
- **articles**
- weblogs
- about

## Managing Multicultural Projects with Complementary Practices

© 2001 Johanna Rothman

This article originally appeared in Cutter IT Journal, April 2001

### Introduction

Any project where people bring different assumptions about working norms (either in product development or team behavior) is a multicultural project. Even when all project participants are from one country, the project manager (PM) may still have to deal with multiple cultures and those cultures' expectations and behaviors.

Some of the team differences are strictly cultural, while others stem from varied management styles and strategies, but all these differences will eventually show up during the project. Some project managers try to manage those differences by mandating common practices and techniques across the project. Instead of trying to get everyone to work in lock step, I've found it useful to get people to agree to major milestones and what they mean and to use complementary practices to achieve project success.

Complementary practices are those techniques and agreements that help each team get their jobs done. For example, one team might perform nightly builds and smoke tests. Another team might do weekly builds and full regression tests. These practices are complementary—they are not the same—but the practices work to achieve the same result: easy integration of multiple teams' work. Each team determines what makes the most sense for them

Google™ Custom Search

Newsletters

Enter your email to subscribe to  
*The Pragmatic Manager:*

[Read previous issues.](#)

Services

[Project/Process Assessments \[→\]](#)

[Management Consulting \[→\]](#)

[Project Jump Start \[→\]](#)

[Management Coaching \[→\]](#)

[Workshops \[→\]](#)

[AYE Conference \[→\]](#)

Articles

[Project Management \[→\]](#)

[Management \[→\]](#)

[Hiring \[→\]](#)

[Requirements \[→\]](#)

[Consulting \[→\]](#)

to do, as long as they meet their agreed-upon deliverables.

## Problems of Multicultural Projects

In my experience, any strictly technical (product) problem solving is secondary to managing the people interaction issues. When managing multicultural projects, I expect problems in these areas:

- **Different definitions of milestone and handoffs**, leading project teams to misunderstand their commitments and handoffs to other groups. Sometimes the different definitions are due to a lack of understanding of the actual words. Sometime people differ on their meaning of commitment- is a commitment a best effort agreement or will the team do whatever it takes to meet the commitment? Whatever the cause, different meanings for milestones can be overcome with complementary product development practices, especially in project planning, project scheduling, and technical review.
- **Uneven project communications and reporting of project state**. This frequently leads to lack of trust in other teams. If you don't know what other people are doing, you may not feel you can trust them. Especially when geography and culture separate teams, this lack of trust can be a huge obstacle to project success.

You might know what other people are doing, but you might not know what they are saying. Language differences, and everyone's relative ability to use one common language can create many problems in a project. What language are you using as the default language? What kinds of ambiguities do you have in that language? How fluent are all the project participants in the project's language? Is everyone willing to talk to everyone else, or are there cultural mores that make some people uncomfortable talking to certain teams or team members? Make sure that the language you use for written and spoken communications is adequate for everyone.

There can also be communications problems with regard to holidays, vacations, and overtime. Be specific about what vacations mean, the impact of everyone's national holidays on the project's schedule, and general expectations about overtime - these will all affect how the project participants work with each other and report on project state.

- **Uneven ability to use common tools, including the project's intranet**. This leads to inability to share designs, source code, tests, and any other project information. When some members of the project team can't use the project resources, they may resent the people who can use the project resources. In addition, they may stop trying to share their work with the rest of the project team.

To have a chance of success, the PM of a multicultural project

[Process Improvement and Quality \[→\]](#)

Weblogs

[Managing Product Development \[→\]](#)

[Hiring Technical People \[→\]](#)

Books

[Manage It! Your Guide to Modern, Project Management \[→\]](#)

[Behind Closed Doors: Secrets of Great Management \[→\]](#)

[Hiring the Best Knowledge Workers, Techies & Nerds \[→\]](#)

[Corrective Action for the Software Industry \[→\]](#)

[Amplifying Your Effectiveness \[→\]](#)

About RCGI

[Speaking Calendar \[→\]](#)

[Testimonials \[→\]](#)

[Resume \[→\]](#)

[Contact \[→\]](#)

must address these problem, for himself or herself and for the rest of the project staff.

## Define the Milestones and Handoffs

Development practices don't have to identical for every group, but the outputs of each group must match the expectations of the other groups. This matching of expectations reflects complementary practices among the project groups.

The first thing I do as a project manager on a multicultural project is to agree on the meaning of important terms and milestones.

### Terms

Project terms vary from team to team as much as practices do. Many teams have their own interpretations of terms such as fix, verify, feature freeze, code complete. You don't have to be part of a multilanguage team to have trouble with terms.

I once managed a second-line support group for a globally dispersed team, in which there was some confusion about the term "fixed". The job of the Boston-based group was to fix the defects that the first-line support group could not fix and that were time-critical for our customers. We had a recurring problem with two of our European first-line support groups. The Europeans repeatedly promised imminent fixes to very high profile customers, because they thought the defects were fixed. However, the defect fixes were not complete. The Boston group was using the notation "Fix" for defects that had been investigated, the cause known, and a fix was in test. "Verify" was the notation for completed fixes. It never occurred to our European counterparts that "Fix" was not truly fixed.

Other terms some teams have found confusing are any project milestones containing the words "freeze" or "complete", such as feature freeze, code freeze, and code complete. I once worked on a project where the US developers thought code complete was the first time they froze the code to create a build. The Russian developers thought code complete was the last freeze to create the final build to generate the master production CD. The technical leads kept arguing during schedule development, until they realized they weren't using the same terms.

### Milestones

Not only do we have to agree on our project's terms and milestones; we need to make sure everyone agrees on what the milestones *mean*. Many years ago, as a program manager, trying to bring together project components from Boston, Los Angeles, and Japan. The technical leads and I were working on the schedule. Everything was smooth until we tried to agree on the first milestone: feature freeze. To the Boston team, feature

freeze meant that the low-level design was complete. However, to the Los Angeles team, feature freeze meant that they had a good idea of the high level design. The Los Angeles group couldn't understand why Boston would want complete module interface designs—Los Angeles wanted maximum flexibility to add features to the product as late as possible. The Boston and Japan teams wanted to define the features early, and freeze the interfaces as early as possible, to allow for the customization of the product for Japan.

I brought the technical leads together to talk about what each group needed and when. Initially, the Japanese technical lead was reticent to express his views, concerned that he was pushing his perspective on the entire team. We revisited the project requirements: release the English and Japanese versions of the product within the same calendar month, and create a public API for the English language market. We didn't have time to retrofit features in for the Japanese market.

Instead of pushing for a resolution, I asked each team lead to talk about their problems and what would create a solution for their problems. The Boston team needed to freeze the API in time for the Japan team to develop their customizations and for the writers to document the product. The Los Angeles team needed to create enough product infrastructure that they wouldn't have to change the API for the next release. The Japanese team needed to modify the GUI and the data structures for the Japanese market. The later Boston and Japan defined the features, the harder their jobs were. The earlier the Los Angeles team defined the features, the harder their job was. Once we realized that we were all on the same project, but we had different goals, we were able to better articulate what we wanted at which time.

As a project team, we were able to develop our major milestones together, by focusing on interim results (what did each group absolutely need by when?). It took us about a week to come to a consensus about what each milestone meant, especially "freeze" milestones, and how we knew we'd met those milestones. Not everyone liked the whole schedule, but we could all live with it.

I call this technique of defining milestones by the results you want "discuss and publish". Some teams chose to define interim milestones in addition to the milestones defined in the overall project plan. When the teams agree on what each project milestone means, you can develop a joint project schedule, and understand what you have to do achieve those milestones.

### **Milestone Achievement**

When managing multicultural projects, I don't mandate how each team should work to achieve their deliverables. Already established teams generally have some built-in practices, including work product review, configuration management, product build, product test, and others. For example, although I

strongly believe in nightly builds and smoke tests, I don't demand each project team perform nightly builds. Instead, I focus on defining and managing milestone criteria, to know that each team has achieved what it said it would, when it agreed to.

## Reviews

After we agree on the milestones, I broach the subject of technical reviews. All projects benefit from reviews, but I've found that on multicultural projects, technical reviews provide an additional communications framework and a context in which to discuss the project issues. Some people are uncomfortable talking about the project-wide issues. Those people may be less reluctant to discuss the technical side of the project, and requirements and architecture reviews provide a framework for them to air issues. I always schedule requirements and architecture reviews for multicultural projects, and I encourage technical review of other documents. I find that by initiating technical reviews at the beginning of the project, the individual project groups are more likely to continue with technical reviews for their pieces of the project.

Multicultural projects require more formal requirements and architecture reviews than other projects; the formalism helps reduce the risk of communications problems. Some people may not comment except in a formal review mechanism—some people may not realize you *want* them to comment unless you have a formal review mechanism. Even if people are willing to comment on requirements and architecture, if you don't make time in the project, the people from various teams will not be able to comment. After all, on an international project, you're not going to run into each other in the cafeteria.

I use formal requirements and architectural reviews to ensure that everyone on the project understands the project objectives. The formal reviews should include at least one technical representative from each project team. These participants agree that the requirements are correct and can be implemented by the team.

In my experience, e-mail-only reviews and inspections are not adequate for effective review of requirements and architecture documents for several reasons [1]:

- The people who first read the work product "direct" the discussion. Some people are too shy to bring up their issues electronically.
- Some people don't read the product once other people start commenting.
- It is difficult to get people to agree on a consistent commenting style.

The cultural differences and therefore the focus of the discussion cannot be bridged without some audio contact. I prefer face-to-

face discussions, but when that's not practical, conference calls may be adequate. In my experience, the way people use and understand language to write specs and their comments tends to prevent effective e-mail reviews. This is especially true when most of the project teams are native English speakers and a minority are not native English speakers. I prefer to get the technical people together in person to review requirements and architecture documents. I find that the travel cost is significantly less than the potential risk of product failure.

If you're working on a very short project, and are willing to take the risk of inadequately meeting the needs of some potential customers, consider some of the Internet-based tools for meetings, coupled with an excellent audio connection. Make sure the moderator is a skilled meeting facilitator, especially when it comes to conference calls. One of my clients uses a checklist for effective conference calls (see [sidebar](#) "Conference Call Dos and Don'ts for Multicultural Project Meetings").

### Communications across the project

Communication barriers, including language differences and differences in topics that can/cannot be openly discussed, make it more difficult to develop complementary practices. However, I've found that the work of defining milestones and terms helps make those communications barriers more obvious.

Once the communications issues are obvious, you can start fixing them. One of the frequent communications problems is understanding how local cultures play- into what gets said and what is understood by the project teams.

On one project with teams based in Boston and Germany, the Boston-based product development team was dependent on a German subsidiary for an independent, but necessary part of the software product. During project planning, the German project lead had repeatedly told the Boston project lead that the German August vacation schedule effectively meant that no one was working on the project for the first three weeks of August. The original schedule planned the product ship date in mid-July. Sure enough, the schedule slipped. The German technical lead explained that no one from the German team would be able to help with product integration during August. They may as well slip the project from mid-August to mid-September. The German lead was aghast that the Boston lead expected the German team to give up their vacations:

*"Why should we give up our vacations because they can't keep to a schedule? We can't take vacation four weeks later- our children are all in school then. Postponing vacation is not an option, and I repeatedly told him that. Why didn't he make his team meet their dates?"*

The PM must understand what is important to all the people on the project. One of my colleagues said about multi-geographical projects: "In order for it to work, it takes a lot of work, like a marriage." [2] In this case, the Boston project leader did not understand the importance of inviolate vacation planning to the German team. I've noticed that many European teams keep their vacations inviolate, while most US teams feel free to change their vacation plans.

This communications problem had two components: the first problem was the project manager understanding that each group had different cultural norms around accepted work behavior. The second problem was how each team understood the actual schedule and the actions the entire project team was going to take if the project slipped.

PMs of multicultural projects should expect that each team has different expectations of acceptable work behavior. Here, the Germans were not willing to change vacation plans. Other European cultures never expect to work overtime. In some cultures, it's unacceptable to question a manager, especially if the manager is a different gender. I once taught a class to Japanese programmers who were uncomfortable asking me questions; they were concerned that I would feel offended. One of my male colleagues has female Asian staff members who won't ask him questions, especially if those questions look like they are questioning authority. However, his staff will ask other women in the organization. n. As a PM, your job is to open the communications bandwidth by considering other group's behavior. As you work with different people and groups of people, you may be able to change how they behave, but they're not going to change just to make it easier for you.

One way to deal with the schedule issues is to publish project plans widely and report on progress frequently. Using inch-pebbles for the last few weeks before critical milestones, such as project shipment is one technique to manage the risk of project slippage [see footnote 1]. Another technique is to produce an overall project status report weekly. To prevent project status reporting from becoming a huge burden and useless, I generally republish just the planned and actual major milestones and intermediate milestones to the entire project team.

## Building trust

Another common communications barrier is people's inability to talk about something important. Some cultures have taboos against discussing attrition, schedule slippage, or other potential risks or project problems. One way to deal with taboo subjects is to create an atmosphere of safety in the project. I've done this with anonymous email, with project suggestion boxes, and visits to each team on the project.

During each visit, I make sure the project team knows that they

can tell me about any problem or risk, without making me angry. I explain at the beginning of the project that I want to avoid surprises, not bad news. I make sure that my PM behavior is consistent, so the entire project team gets the message that they need to tell me problems up front. I trust the project teams to tell me what their concerns are, and they trust me to help find a solution, not yell at them.

On a recent project with groups in Israel, Boston, and Chicago, the PM was concerned that although everything sounded great in the project, something was wrong. The PM didn't have any specific information, just a gut feel that something wasn't being discussed, and therefore was festering. The group leaders were all together for a meeting, and I broached the subject by asking, "What's the dead fish on the table?" The three group leaders and the PM were surprised by my question. "What dead fish?" I explained that there was something they weren't talking about. If we didn't talk about it soon, we weren't going to be able to do anything about it. The Israeli lead jumped up, and said, "I'll tell you what the dead fish is. You Americans are going to get wonderful bonuses, and we Israelis aren't!" The rest of us sat there, completely surprised, and the PM said, "Says who?" The Israelis had heard a rumor, and instead of verifying it, had decided the rumor was true. Now that the PM knew about the rumor, he could address the issue. (No special bonuses were planned for anyone on the project.)

PMs can use several techniques to help establish trust, including asking the project group what rumors they've heard. Although people may not speak up at first, by asking this question at every meeting, the group will come to understand that you genuinely want to hear about potential problems and concerns. Other techniques, such as checking on how "safe" people feel in the meeting by anonymous vote (i.e. on a scale of one to five, write down how comfortable you feel expressing your honest opinion and hand in the paper) [3] and by specifically looking for risks, such as schedule chicken [see footnote 2]. [4]

## Using Common Tools

What's the point of defining and buying common tools if they're not common? The entire project needs to use a common defect tracking system, and configuration management system. Make sure you have an intranet that everyone can access at equal speeds, so the entire project team uses it. Make sure any tools you have are easy to use over the network and have adequate response time.

Sometimes, the tools appear to require people to track defects, build, or to track requirements in a specific way. Every tool I've used has been sufficiently flexible to allow various uses. When the project staff tells me the tool is defining their process, and they don't like that definition, I ask them to tell me what they do want. I help verify the problem they want to solve is the same



problem I want them to solve. Once I understand the problem they're trying to solve, we jointly determine how to make the tool do that.

One way to make sure everyone understands how to use the tools is to have people in different groups visit one another to show how they use the tool back at their office. The in-person visit helps people see new ways to use the tools and also how others in the project use them. Such sharing builds camaraderie as well as reduce some jealousy that might develop as one group learns that another really knows how to use a tool that they do not know how to use.

I run into the tools problems most frequently with defect-tracking and configuration management tools. On one project, the Boston team used the defect tracking system to track everything they had to get done after the "initial code complete" handoff to formal system test. The San Francisco team only wanted to put problems with known solutions into the system, and the German team wanted to put only true defects into the system. I asked the team leads to discuss the problems they were each trying to solve. I wanted to track defects the same way over the whole project, and I could have lived with either the Boston or German team's solution (the San Francisco team's solution was not adequate), and after I explained why, the groups jointly agreed on a solution.

## Summary

Not all teams on a multicultural project have to do the same things at the same time. However, all multicultural project teams have to agree on major milestones, and the criteria for those milestones. The team as a whole also needs to agree on what to do if the project runs into trouble, and release date has to slip.

Make sure each project group knows what they need to do, and when, but leave the "hows" up to each group. Communicate with the group as often as possible, and make sure the groups know they can talk to you when they have problems.

Require the minimum set of tools, such as defect tracking and configuration management. Decide how important it is that each group use the same tool. I've generally insisted on a common configuration management tool, but that's generally the only tool I insist on. Make sure required tools are easily usable by each project group.

Multicultural projects are challenging and fun. It's hard work and takes focus on the people issues. When the projects succeed, you have made something more than the sum of the parts.

## References

1. Rothman, Johanna. [Six Tips for Making Global Teams Work](#),

Software Development, August 1998.

2. Purchia, Barbara. Private communication. 1995.

3. Smith, Steve. "Safety Check", <http://www.stevenmsmith.com/>, 2001

4. Rothman, Johanna. Ending Schedule Chicken, *Reflections*, Volume 3, #3. November 2000.

## Acknowledgements

I thank these people for reviewing early drafts of this paper: Esther Derby, Jon Hagar, Dwayne Phillips, Barbara Purchia, Bertrand Salle, and Steve Smith.

## Footnotes

1. For more information, see [How to Use Inch-Pebbles When You Think You Can't](#).

2. Schedule chicken occurs when everyone knows that they're not going to meet their part of the schedule, but rather than admitting it, they wait to see if someone else will speak up first.

---

## Sidebar: Conference Call Do's And Don'ts for Multicultural Project Meetings

**Basis Technology**, an internationalization/localization company, works on only multicultural projects. They've developed this set of conference call guidelines.

## General Facilitation Guidelines

- Introductions: as all sides to announce who on the call and who they are (their role).
- All participants should agree that facilitation is necessary.
- Use the "one conversation at a time" rule.
- Say who you are when you speak.
- Say the name of the person if you are addressing someone in particular.
- No eating (especially no eating something in noisy packaging)
- If you are interpreting to another language lay down the ground rules (interpret after each complete thought) and don't be afraid to enforce them.

## Logistics

- Mute button (make sure you don't have music when you

mute)

- Check the mute button is really muting you.
- Use a good speakerphone for full duplex sound.
- Get everyone on the calling side together before calling the other groups.

## Meeting Planning

- If not everyone is there after 15 minutes, reschedule.
- Give participants a cell phone number to reach you in addition to the conference call number.
- Publish an agenda that everyone can see - this is even more critical than in a face-to-face meeting.
- Keep topics focused, get the right people to discuss those topics
- Make sure the meeting objective is clear.
- Vary the order of participants (who speaks first etc.) especially for regularly scheduled calls.
- If there is no good reason to meet, cancel the meeting.
- Have an end time.
- Use collaboration software, such as NetMeeting, when sensible.

## Meeting Facilitation

- If someone joins the call late have someone else, not the facilitator, summarize what's happened so far and say who is on the call.
- Use mirroring (repeating what the person said)
- Use focused conversations (See R. Brian Stanfield, ed. [The Art of Focused Conversation, 100 Ways to access Group Wisdom in the Workplace](#). New Society Pub. April 2000)
- Know when to cut off specific questions.
- Ask for people who do not agree (i.e., "Does anyone not agree or not understand?").
- Be sensitive to when other people have lost the conversation.
- Confirm completion of each topic, organize things topic by topic.
- Be aware that it is much harder to understand a non-native speaker over the phone
- Check that people have not been cut off in the middle of the conference call.
- Summarize partway through and at the end.
- List action items at the end of call.
- Announce when the meeting is over.

## After the teleconference

- Always send a summary of teleconference with action items to all participants as followup.
- Make sure you know when the call has really ended (is speakerphone still on?)

---

Johanna Rothman observes and consults on managing high technology product development, looking for the leverage points that will make her clients more successful. You can reach her at [jr@jrothman.com](mailto:jr@jrothman.com) or by visiting <http://www.jrothman.com/>.

---

What can we do for you? Go to: *Writings and Presentations*,  
*Chronological Listing*

[home](#) • [services](#) • [articles](#) • [links](#) • [books](#) • [templates](#) • [site map](#) • [contact](#)

*Copyright © 1998-2007 Rothman Consulting Group, Inc.  
All rights reserved.*

ROTHMAN CONSULTING GROUP, INC.  
Johanna Rothman, President

[jr@jrothman.com](mailto:jr@jrothman.com) || Phone: +1-781-641-4046