

Making Telecommuting Work

Johanna Rothman

Jill, an experienced software developer, came to me one day and said, “JR, since we’ve moved here, my commute is now 45 minutes instead of 15. I’d like to start working from home two days a week.” I voiced my first thoughts, namely, how could we make it work? Jill said, “Well, you trust that I can do the work, right? You know I’ll talk to people when I need to, that I’ll deliver what I say I will, and that I’ll be in for your group meetings. What else do you want?” I said, “Yes, I trust you to do the work and interact with people in the office appropriately, but I don’t know how it’ll work when you’re at home. How do I know you’re going to work when you say you will? I still need you to work during the workday. I need



to know you’re not going to slack off, but I also need to know you’re not going to work all hours of the day and night and burn yourself out.” We talked more and came up with an arrangement that’s still working for Jill, although I’m no longer her manager.

Many people predict that we will all work from home either part time or full time in the future—we’ll be “excellent engineers” (T. Peters, “What Will We Do for Work?” *Time*, 22 May 2000). Telecommuting might be the wave of the future, but many of us are concerned that we can’t create the kind of team we want to have on our projects, that telecommuting will mean out of sight, out of mind, or that we won’t know what our telecommuters are doing. Software

development is especially difficult to manage when everyone isn’t around all the time. We have to talk to each other to design, review each other’s work, and understand what’s happening in the project.

I’m not going to talk about the legal issues of creating a home office here. There might be Occupational Safety, Health, and Administration requirements (in the USA) or other local and national requirements for an employee’s home office—you can investigate those requirements with your local authorities. I’m going to address how to make telecommuting work for you as a software manager.

It *is* possible to create a telecommuting culture—an environment in which people can successfully work from wherever they happen to be physically. The critical component of that culture is building and maintaining an open and trustful communication between the manager and employee and between the employee and the rest of the team. One way to build that three-way communication is to create a track record of successful deliverables by clarifying the work required, facilitating interaction rules, clarifying how people work with each other one on one, and specifying how the telecommuter participates with the team.

Create a Track Record of Successful Deliverables

A successful deliverable is the act of completing work on time in a way that fits the rest of the team and the project. A key piece in creating successful deliverables is to clarify everyone’s expectations of what the de-

liverables are, when they will be delivered, and the role the telecommuter plays in delivering that piece.

Jill was a senior software developer. Her job was to refine the current product architecture as we added new components to the product. She created high-level and some low-level designs, she reviewed other people's low-level designs and sometimes moderated design reviews, she implemented the code for her designs, she tested what she had done, and she mentored junior team members.

We did three things to make sure her deliverables were successful: we clarified Jill's role on the project and on the team; planned our interaction rules—the ways we would communicate with each other; and discussed how she would participate with the team and with me, her manager.

Specify the Work to Do

When you don't have many easy opportunities to check in with your staff, it's especially critical to explicitly define the employee's tasks. Some managers do this with individual task lists or with a project's work breakdown structure (WBS), especially using inch-pebbles (one- to two-day tasks). Because Jill was a senior person and I had created accessible project schedules with WBSs, we could work the way we always had. We talked first about what she was going to do, and then she developed her schedule. I fit her work into the project schedule, and she reviewed it (along with the other tasks) to make sure things fit from a design perspective. We were used to leaving the schedule accessible to the entire team, so Jill could continue doing that from home.

If you're working with an external project participant, clarify the work that person must do and clarify any rules you have about how the work is done. One of my colleagues, Brian Marick, regularly participates in pro-



jects as a long-distance test developer. He works with the project manager and the project team to define the overall approach to the work, any specific deliverables the team requires, and how they want him to participate with the team.

Specify Interaction Rules

We discussed how Jill would work—we didn't discuss how Jill would design or code, but we did discuss how she would work with the team. When would she be on site and for how long? How was she going to make sure the designs were reviewed? How was she going to moderate design reviews? What decisions would she make alone, what decisions would she make with the team? What would trigger my management involvement?

At first, Jill laughed at me and called me a worrywart. Then she agreed that working away from the group might require us to write things down in a way we hadn't needed to before. She generated her first pass of interaction rules, and sent it out for review.

Peter, one of the junior developers asked, "When does Jill ask for help? When do I get to ask her for help?" Jill didn't need help in the same way a junior employee did, but she did need to recognize when she was stuck, just

like the rest of us do. Jill and Peter talked, and Jill came up with the 45-minute rule. If she were stuck on something for 45 minutes, it was time to talk to someone else. She had to take the initiative to notice if she were struggling, no one else could.

Jill was not fond of interruptions, and she often put a sign on her cubicle that said, "Thinking, please wait to disturb." The sign worked well as long as she was physically visible. How would anyone see that sign now?

Jill decided to buy a phone she could forward to voicemail without hearing the ring, and she promised

to check her voicemail when she finished her heavy-concentration work. She also decided to measure how long she spent in think mode, so she could predict how long it might take for her to get back to someone.

Specify Participation with the Team and Manager

When you create a project team with people who are not always physically present, you might need to organize the team differently. I was concerned that our group meetings—primarily problem-solving sessions—would be difficult to run with Jill present only on speakerphone. I told Jill this and asked what she would do to make sure she participated fully.

Jill came back to me with some physical requirements (speakerphone quality) and with some meeting management requirements—because Jill was separated by distance, we needed time to let her get a word in edgewise.

I decided to become a better facilitator of geographically distributed meetings and practiced those facilitation skills for a variety of meetings. (I created a checklist for myself, using some of the ideas in the *Facilitator's Guide to Participatory Decision-Making* [S. Kaner et al., New Society Publishers, 1996].)

I was also concerned about how we would meet one on one during the

My Checklist for Facilitating Meetings with Remote Employees

1. Pause. Wait to make sure everyone has a chance to talk.
2. During each discussion point, ask the remote staff if they have questions or comments.
3. Summarize each action item and outcome of each discussion point.
4. Stack people who want to talk, checking with the remote staff every couple of people, in case they're trying to get a word in edgewise.
5. Make sure people understand your sense of humor and other people's jokes. (Sarcasm doesn't easily translate over the phone.)

week. I use employee one-on-one time to discover the technical and nontechnical issues the employee has wrestled with during the past week and any progress made. For me, it's even more important to touch base with a remote or telecommuting employee on a weekly basis. Jill and I set aside time each week to discuss what she was working on. At first, those days were days she was in the office. Jill then wanted to use her office time for reviews and inspections, so we successfully moved our one-on-one time to a time she was in her home office.

Contra-Indications for Telecommuting

I'm concerned with someone's capability to telecommute if they have trouble asking for help or if the employee has trouble interacting with others at work. In each case where I've had trouble making telecommuting work, the telecommuter either wasn't able to ask for help in a reasonable amount of time or was unable to fully interact with other project people. Junior staff or part-time staff can be more vulnerable to these contra-indications.

Asking for Help

It can be scary to ask for help. Some of us would rather keep working at something, because we know we'll get there eventually. In this group, we'd already established a culture that emphasized asking for help as good and expected. As a manager, I made a practice of reviewing estimates and actual plans

so we could become better estimators. I try to keep a balance between letting people investigate and try new things and the schedules we had to meet.

If your group doesn't have the ability to easily ask for help as part of its culture, consider how you will overcome your engineers' natural reluctance. Your telecommuters trust you to assess their performance wisely and not to downgrade their performance because they asked for assistance.

Interact Fully with Others

In my experience, new project engineers might be reluctant to get up-front coaching or mentoring from other project team members. Many of us prefer to explore the product on our own; we don't necessarily want to be told how things work. Although this is a reasonable strategy for learning about some products, it's not appropriate for all products. Many times, especially on large or complex products, it's more appropriate to learn a little about the product on your own and then test that learning with other people on the project team. If one team member doesn't like checking things out with others, that team member is probably not a good candidate for telecommuting. Successful telecommuters make the effort to reach out to the rest of the project team, to learn what the team knows, and to share what they know.

The telecommuter has to understand what it's like to work in your

environment and must be able to adapt to your environment. People at a distance do not shape the corporate culture; the people who are there every day shape it.

Making telecommuting work is like a requirements problem (B. Lawrence, "Requirements Happen...", *Cutter IT J.*, Vol. 10, No. 4, April 1997, p. 3). Who are the people the telecommuter affects? What attributes does the telecommuter need to have? What attributes does the group need to have? How will you manage the risks?

First, build trust between the telecommuter, the manager, and the project team. As the manager, you do this by specifying the work you need done and by creating the ways in which your project team communicates with each other. You can decide how the team works but facilitating the decisions about how the team will work together is even better. Notice when the telecommuter is in trouble. Not being able to ask for help is a common problem, but not being able to start or finish the work are too. Make sure you and the telecommuter have discussed how you're going to avoid problems and recognize them when they do arise.

As managers, we're always looking for ways to retain our valued staff. Telecommuting can be one perk in your arsenal of employee retention tools. ☺

Acknowledgments

I thank the following reviewers for their valuable comments: James Bullock, Erwin van der Bij, Danny Faight, Karen Mackey, Walden Mathews, Brian Marick, and Pat McGee.

Johanna Rothman observes and consults on managing high-technology product development. She works with her clients to find the leverage points that will increase their effectiveness as organizations and as managers, helping them ship the right product at the right time and recruit and retain the best people. She's published a handbook for perplexed managers and articles in *Software Development*, *Cutter IT*, *Computer*, *Software Testing and Quality Engineering*, *Crosstalk*, and *IEEE Software*. Contact her at jr@jrothman.com or www.jrothman.com.