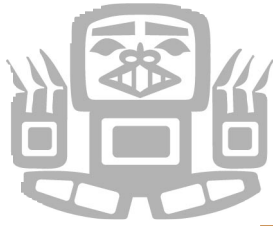




Manager



Roger Pressman

Fear of Trying: The Plight of Rookie Project Managers

Recently, a front-page article in *The Wall Street Journal* (4 April 1997) observed that "...many [technical] people don't want to be managers, and many people who are managers are, frankly, itching to jump off the management track—or already have." Describing a phenomenon that it called "management phobia," the article noted that sentiment against moving into management positions is the highest it's been in more than two decades. The article profiled young technologists, who in another era would have moved gladly toward project management, but today shun such positions. Most cited the "Dilbert Factor" as their primary reason for staying put.

Little more than a month later, *The Wall Street Journal* (14 May 1997) featured another front-page article—"A Software Engineer Becomes a Manager, With Many Regrets"—that addressed this topic from a different angle. A talented young software engineer working for an aggressive software company was promoted to lead a five-person project team. The team struggled to build a business-critical application under a tight deadline, only to have the product ship six months late. After the dust settled, the young man decided to give up his management position, saying, "Within three weeks, I'll just be spending all my time coding. I'll never have to do a review [personnel appraisal] again."

As a software engineering consultant, I've noticed this phenomenon among the ranks of software engineers in companies large and small. Most don't want the hassle of project management, feel (and often are) ill-prepared to take on the responsibility, and abhor the politics and frustrations endemic to the position.

And yet, few senior managers would argue with the notion that poor project management is the number one cause of project failure. We need good project managers, but it seems that the best and the brightest practitioners have a fear of trying the project management route. What to do?

Hundreds of books and training courses and dozens of management theories address software project management. Each discusses the techniques and tools that lead to a successful project and the attributes exhibited by successful project managers. After more than 20 years of consulting with software devel-

opment organizations, I've come to believe that successful projects have much more to do with the person leading the team than with project management techniques and tools.

So, how do we grow good project managers? What do we teach the rookies who have just been appointed to lead their first software project?

Regardless of the training or mentoring approach you use, I suggest focusing on four major attributes, which I describe in their order of importance.

COMMUNICATION

Here's the scene. An IS steering committee of senior executives summons a young project manager to report on the status of a troubled, critical project. He begins by looking at his shoes, unfolding a sheet of crumpled paper, fidgeting nervously, and then, without preamble, says "We've run into a roadblock in module TCP/IPxcon. The status bit that should be set by module, uh, I forget it's name, anyway, its a real-time control module in the network management architecture, has, uh, given us a bit of a problem because, well, we thought that the requirements were consistent with...."

I suspect you've been there.

Project managers must understand how to communicate, and more important, how to tune their communication to their audience. It doesn't matter whether the communication is a presentation, a written report, or a phone call. It must be structured in a way that will get the message across clearly and concisely.

Some people have a natural instinct for communication, but most do not. Rookie managers should be trained to express the same idea appropriately to each different constituency. They must understand the needs of their audience and shape their presentations to meet them. On a given day, they may have to deliver the same message to executives, technologists, customers, and end users. The overall thrust of the communication may be the same, but the tone and structure will differ radically for each constituency.

Can this be taught? Yes. Can it be learned on the job? That depends. If rookie managers have competent mentors willing to spend time critiquing and advising on all communication to all constituencies, rookies *can* learn



on the job. But if rookies are thrown into a project with little or no training or help and expected to “understand” how to communicate, problems will result immediately.

NEGOTIATION

Rob Thomsett, a well-known and widely respected consultant in the software project management area, talks about “first, second, and third wave” project management. In the 1960s and 1970s, during the first wave, software people held all the power and dictated delivery dates and costs to end users and customers. In the 1980s, during the second wave, a more balanced power relationship existed, at least in principle: business and software people worked together to derive requirements and mutually set deadlines and costs. Now, during the third wave, the balance of power has shifted to end users and customers. This means that they, not software developers, dictate the rules of the game. It also means that rookie project managers had best learn to negotiate—with their customers, with the technologists on their teams, and with the business executives who oversee their work.

There are many different ways to negotiate, but all of them can be summarized in the five steps every rookie project manager must learn.

1. *Establish a dialogue.* Because software project managers are no longer in a power position (as they were during the first wave), it is critical that they apply the communication skills I described earlier. They must probe, offer their own ideas and suggestions and build on the client's, and constructively criticize requirements that will lead to trouble. Project managers must work to make the customer understand that successful software will magnify the customer's new-found power, and that such software can only be created through a close working relationship.

2. *Plot a negotiating strategy.* Rookie project managers probably believe that quick thinking during discussions with end users and customers is the key to successful negotiation. In reality, it's most important to plan a negotiation strategy in advance, before you make any attempt to overcome obstacles and come to terms. In essence, managers must answer these questions: What will be negotiated? Who are the players? When and where will the meeting take place? Once they know these answers, they can better analyze their position, organize the information, and assess alternative solutions and positions that may arise. The vast majority of rookie managers walk into a customer meeting ill-prepared. They often get steamrolled as a result.

3. *Identify, then overcome, obstacles to success.* For example, while discussing requirements and delivery dates for a major legacy-system enhancement, a rookie project manager and an end user reach an impasse: the customer is insisting on an impossible deadline. What should the manager do? One approach is to initiate a change of pace. In this case, the manager might say something like, “I think we've

made good progress and I want to continue so that we can finalize things and begin this project. Why don't we take a break and then get back together at, say, 3:00 p.m.?” The customer may agree and be more willing to compromise after a break. Or, things may not be so simple. The customer may respond to this offer confrontationally or even irrationally. In cases like this, negotiation training is invaluable.

4. *Come to terms.* Once rookie managers neutralize obstacles, they must actually conduct the negotiation. To do so, they should open with a statement of purpose and review the pertinent information. Managers must recognize that both parties have needs and that these needs can be fulfilled in several different ways. They must work with clients to arrive at the best alternative. Finally, managers must “close” the negotiation. That is, they must summarize the agreement and identify both parties' responsibilities and the steps that follow.

The vast majority of rookie managers walk into a customer meeting ill-prepared.

5. *Make it happen.* Using organization and facilitation skills, software project managers begin the technical aspects of the project, but must never forget that, for the project to succeed, communication and negotiation must be ongoing activities.

The vast majority of rookie software project managers have never received *any* formal training in negotiation. Few have even read a book on the subject. They do not know these five steps and thus cannot apply them when they meet with end users and customers. The result? Misunderstandings, insane deadlines, unclear requirements, and the tension and frustration that lead to management phobia.

ORGANIZATION

Many people bumble their way through life, often just one wrong move from chaos. Amazingly, this approach can work, but not for rookie or even experienced project managers. Managers need organizational skills to administer technical work, coordinate the people who do it, and track and control the resulting products.

Organization is a partitioning process. Managers must know how to partition the work to be done. Both product functionality and the tasks associated with the software process must be partitioned and then related to one another.

Managers must also partition their interactions with project team members. It is difficult to assume the roles of advisor, confessor, parent, cheerleader, and even disciplinarian all at the same time. Attempting to do so invariably leads to chaos. Before entering a meeting,

Continued on page 54



Manager

Continued from page 51

a review, or a one-to-one conference, managers should consider which roles are most important for the given situation.

Finally, managers must understand which work elements require immediate attention and which can proceed without direct supervision. On large projects, it is difficult and inadvisable to track and control all work tasks with equal emphasis. By partitioning work tasks and their relative importance to project goals, managers can prioritize and keep the project on schedule.

FACILITATION

In addition to being a communicator, negotiator, and organizer, a project manager should also be a facilitator. Stated simply, the manager's role should be to make things easy for the people who are doing technical work. In the role of facilitator, managers act as a buffer between the "techies" on the project team and those who fund, track, and control the project.

As team leader, a rookie manager should shield practitioners from the time-consuming burdens of everyday corporate bureaucracy. In software design jargon, managers apply "information hiding," treating the team as an encapsulated object in which data and the functions that manipulate that data are, to some extent at least, hidden from the outside world. Managers serve as an interface to the team. They filter communication with team members, not to keep them

in the dark, but to screen out unnecessary and time-consuming distractions that have little or nothing to do with project success. Managers should minimize bureaucratic record-keeping duties and reporting functions to allow team members maximum time for productive work. Finally, meetings should be structured for effectiveness: Managers should set an agenda in advance, demand advance preparation when it's required, and ensure that records are kept and action items identified.

BOTTOM LINE

Rookie software project managers should be given the opportunity to succeed. Ideally, they should be trained in advance and given the necessary tools and resources to get the management job done. But this is the real world. In far too many cases, upper management throws the rookie into a project armed with little training and fewer resources. When that happens, things can be difficult—for the rookie manager, for the project team, and for the project. But even in this less-than-ideal situation, managers can survive if they remember four key concepts: communication, negotiation, organization, and facilitation. All the rest is detail. ♦

Roger S. Pressman is president of R.S. Pressman & Associates, Inc., a consulting firm that specializes in software engineering training and process improvement strategies. For over 25 years, he has worked as a software engineer, manager, professor, author, and consultant. His books and video series on software engineering are widely used throughout the industry.