Becoming a Great Manager: Five Pragmatic Practices

Esther Derby
Esther Derby Associates, Inc.

Barry Boehm famously said, "Poor management can increase software costs more rapidly than any other factor." If that's true, then we should be attending as much to management practices as we do software development methods. In this article, the author describes five pragmatic practices that will help managers focus on the right work and create an environment for success.

I had lunch the other day with a fellow who had just pulled the kids out of school and left for a new job in a different city.

"What was it about the job that made it so attractive?" I asked.

"It was my boss," he replied.

What kind of manager generated that sort of loyalty? It was not a poor manager; poor managers create the illusion of progress with busyness. They depress productivity and morale. It was not an average manager – one who accomplishes work – but not always the right work. This fellow had a great manager, one he was willing to follow halfway across the country.

I have met and interacted with dozens of managers in software companies and information technology (IT) departments. Those who were successful accomplished goals that contributed to the bottom-line results of the company and developed people. They had sufficient domain knowledge to ask probing questions and understand risk. They understood the technology their teams worked with, even when they did not have the skills to implement the technology, and they all followed a similar set of management practices.

The following may not be practices taught in management schools or written down in software management and project management books. They are the common threads that emerge from my observations of successful managers.

Decide What To Do and What Not To Do

Great managers do not just accomplish work, they accomplish the right work. For companies that are in the software business, it is work that helps the company generate revenue, attract customers, and keep profitable customers. For IT shops, it is work that enables the business to operate effectively and efficiently.

Ask yourself how your company makes money and how your group contributes to the company's success. If your group develops the software that the company sells, the connection will be clear. Sometimes, it is not so clear. If you are in a testing group, you may contribute to business success by providing information

that describes risks and enables good release decisions. If you are in a documentation group, excellent technical manuals will reduce calls to the support group and improve the customer's experience with the product [1].

Once you know how your group contributes to your company's success, you can articulate the mission for your group. For a development group, the mission could be stated: Deliver features that meet our customer's need and provide a return on investment for the company. When I managed a group that wrote software for investment portfolios, we determined our mission was: Provide accurate and up-to-date valuations for our funds.

We all have more than enough work to do; a clear mission helps us prioritize strategically important work and identify work that does not need to be done (at least not now, or not by our group).

I worked with a test manager to identify a mission for his group and prioritize all the work his group was doing. He defined his group's mission: Assess and communicate business and technical risks for the applications we test. As he listed the work his group was doing, one activity stood out: client site pre-sales technical support.

Two years earlier, he had helped the sales group by sending a tester on a sales call when a member of the sales team fell ill. The sales call was successful, and the sales manager continued to ask for testers to accompany the sales team on important accounts.

The test manager realized that while he was building goodwill, it was not in his mission to provide pre-sales support at the customer site. As he examined the work his group was doing (including the site vis-

its), he saw that sending testers on sales calls was *preventing* the group from accomplishing work that would help assess and communicate technical risks on applications under test. He also knew he could not drop the responsibility immediately – without his testers, the sales team would flounder, and sales were the lifeblood of the company. He worked with the sales manager to provide training and transitioned to phone support only for calls.

Deciding what to do and what not to do helps focus efforts on the important work – work that will contribute to the bottom line of the company. Articulating a mission has another benefit: When everyone in your group knows the mission and how the work they do contributes to it, they will be able to make better decisions about their own work every day.

Limit Multitasking

It *seems* like assigning people to multiple projects will ensure all the projects are completed. But contrary to popular belief, multitasking does not speed work, it slows work and delays delivery. Multitasking creates the illusion of progress by creating busyness while robbing people of time and mental cycles. Humans are not particularly good at switching contexts [2]. Gerald M. Weinberg quantifies the amount of time lost in Table 1.

People lose time as they put away Task A and remember where they were on Task B. It takes time to retrieve and review documents or notes related to the task and to re-create a train of thought. Assigning multiple tasks buys time only when there are two tasks and one is clearly the top priority. The person works on the top priority task

Table 1: *Quantification of Time Lost to Multitasking* [3]

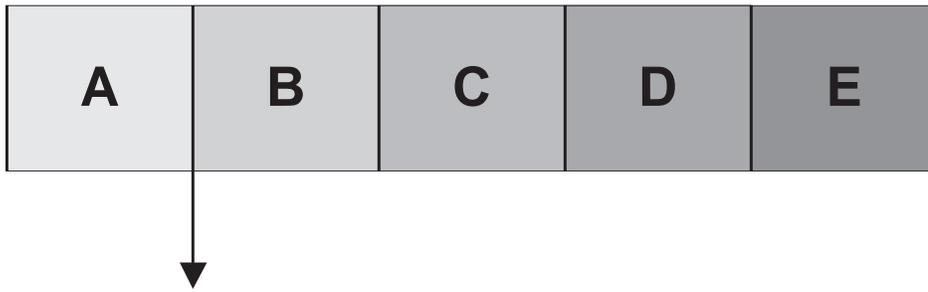| Number of Tasks	Percent of Time Spent on Each Task	Total Task Time
1	100	100
2	40	80
3	20	60
4	10	40
5	5	25

Figure 1: *One Person Assigned One Task at a Time*

until he is stuck or can go no further, then he picks up the lower priority task. This reduces time a person might spend spinning his wheels if he only had one task.

In product development, we can think of tasks as projects, or chunks of significantly different work. Two or more projects zap productivity. Suppose you have five tasks. Figure 1 shows how focusing on one task delivers value sooner. The arrow shows when the most important task, Task A is completed.

What if you have five tasks and only one person to do them? Rather than assign five tasks to each person, create a task queue. As each finishes a task she can sign up for another. This limits multi-tasking and will actually improve delivery time. Assume you have five tasks and they are all assigned to the same person, with instructions to make progress on all the tasks, because they are all important. The person doing the tasks splits her time between all five tasks. Figure 2 illustrates one simple effect of task switching. Each slice of pattern represents time spent on a different task. The down arrow shows the first time the project will earn value from a completed task. (In reality, the delay will be much longer because of time lost with each switch between tasks.)

Assume another project had identical tasks, but the worker focused on one task at a time until it was completed. The project would realize value much earlier, as illustrated in Figure 1. By the time the worker has completed one task, the focused worker has completed three tasks: A, B, and C. Plus, she has started on D [4].

In spite of the evidence, I often see people who are assigned to three or more projects. Managers expect people to spend a certain percent of their time on each project. In reality, the requirements of most projects do not fit neatly into 25 percent, 40 percent, or 75 percent of a week. The worst case is expecting people to switch between every assigned project every day.

In most workplaces, people are not actually working on project tasks a full eight hours a day. In a talk at the Software Development Best Practices Conference and Expo in 2003, Noopur Davis of the Software Engineering Institute reported that across 200 projects, the average amount of time spent on task (work that is directly referenced in a project plan) is 15 hours a week [5]. That is three hours a day! How much can a person really accomplish in 45 minutes – 25 percent of three hours – in each day? The largest slice becomes the de facto priority and everything else is squeezed and rushed into overtime hours when productivity is lowest.

Great managers realize this and assign work (or have people sign up for work) on one project or on two roughly similar tasks. Great managers realize that they will deliver value to the business faster if they finish one important project before starting on another.

Keep People Informed

People work best when they have the information they need. Great managers share what they know about the task and how the task fits into the goal of the project or organization.

People need information about their tasks, but they also need information about the big picture. Poor managers parcel out knowledge on a need-to-know basis. Good managers provide information about specific tasks and also explain how the work fits into the group's mission and the company's success. Being able to put a task into a bigger context provides motivation and helps people make better decisions.

I met one manager who kept an overall project plan to himself and only allowed people to see their assigned tasks. The people on the team had to go behind his back to see who was dependent on their work, who they needed input from, and where they needed to work together. This manager lost an opportunity for people to see their work in a larger context and find efficiencies and possibilities for collaboration. His actions communicated that he did not trust people and viewed them as *biological code producing units* rather than intelligent, creative people who were capable of planning and managing their own work.

People need information about their work. They also need information about events and priorities within their organization. When people lack information, they fill in the gaps with their worst fears and rumors. Rumor and conjecture consume an enormous amount of time in organizations. Communicate what you can, without revealing confidential personnel information or violating clearances or contracts. When you do not know something, admit it and communicate when you will have more information.

Managers need to be informed, too, about status, obstacles, and concerns. Serial status meetings (the kind where each staff member reports status one-by-one to their manager) save the manager time but wastes everyone else's. Typically, they only provide a fraction of the information a manager needs to know. Many important issues remain unspoken in this sort of status meeting. For example, people may be reluctant to admit they are struggling, reveal obstacles, or talk about their professional aspirations in a group meeting. That is information that a great manager needs to know. Hold regular one-on-one meetings to learn about the personal side of work tasks and stay in touch with professional goals. Save team meetings for interdependent work, group problem-solving, and team decision-making.

Provide Feedback

People need to know how they are doing at work. Do not assume that people know when they are missing the mark or what they are doing well. Great managers meet one-on-one with people regularly, every week or every other week to have the opportunity to give feedback when course corrections are small. One-on-one meetings also provide the opportunity to notice and appreciate something about each individual's contribution.

Make feedback specific, so people can act on it. I talked to one woman whose manager told her she was too nice. She was at a loss for what to do. Useful feedback describes behavior or results and states the impact – whether you want the person to change or continue a behavior.

I employ the following structure for giving effective feedback:
1. Create an opening.
2. Describe the behavior or result.
3. State the impact.
4. Make a request or engage in joint problem-solving.

The following is an example of what a great manager might have told the woman who was too nice:
1. I have some observations that I think

might be helpful to you. Is this a good time to talk?

2. I noticed in our meeting today that you agreed to all the requests from marketing for feature changes.
3. I know you are working extra hours and have a long backlog. What I see happening is that you are losing credibility with marketing because you agree to do work that will not be accomplished in the timeframe that marketing expects.
4. Let us talk about what we can do to manage expectations and do the most important work without burning you out.

Great managers do not wait until the yearly performance review to give feedback – that is a prescription for broken trust. If you want someone to be successful, tell him or her what needs to change as soon as feasible so that person can make adjustments. There is no sense in allowing sub-par performance or mistakes to continue; that just drives down productivity and morale for the individual and the team.

Develop People

Great managers know the career and professional aspirations of the people they work with, and they strive to help them meet those goals when they can and help people move on when they cannot.

A longtime employee wanted to move from testing to development. She took on as many technical testing tasks as she could and studied programming on her own time. Because she knew how to read code, she was an asset to the team in understanding how to design tests that did not depend on a graphical user interface. When she told her manager she wanted to apply for a job opening in the development team, he blocked her. "You are too valuable here," he said. The employee was gone within a month, taking a job as a developer at another company, and her team and her company lost a valuable employee.

Great managers talk to people about the skills they want to develop and the direction they want to go. They find opportunities to build skills and capabilities in the day-to-day work. The intention to develop a new skill almost always falls victim to a pressing deadline unless those intentions are planned into daily work. Finding development opportunities in daily work plans builds skill and loyalty. Even when people move on, they remember the managers who invested in them instead of just using their labor.

Another tester wanted to move into project management. The deadline for the project meant it was not feasible to send the tester to an off-site seminar. But
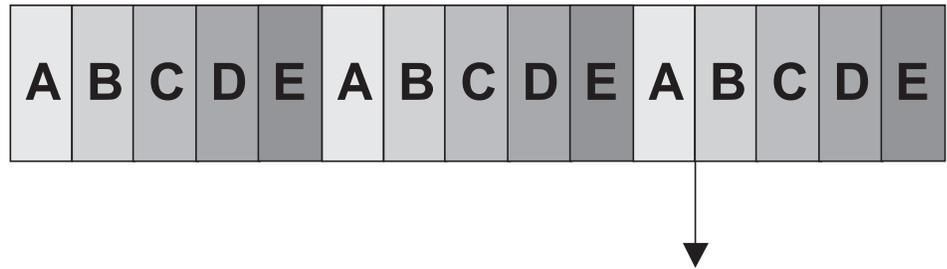

Figure 2: *One Person Assigned Five Tasks With Instructions to Make Progress on All of Them*

together, he and his manager identified a small library of books and articles for self-study. His manager helped him apply his *book learning* as the tester organized and tracked his own work as a mini-project. He coached the tester to break tasks into one- or two-day chunks and define completion criteria. He showed the nascent project manager how to gauge his progress and think through his options when his rate of progress did not match his desired rate. By the end of the project, both the tester and the manager felt confident in moving the tester into a test lead role where he could organize and track his own work and that of two colleagues.

Conversely, when great managers do not have the work that will build skills, they tell people directly. Years ago, I worked with a young man who wanted to move from mainframe programming to PC programming. I knew that the work in our group would not help him reach his goals. Rather than try to hold on to him, I made a few introductions within the company. Later, after he had moved to a different department where he could do the work he wanted to do, he asked me out to lunch.

"You were honest with me," he said. "I felt bad about leaving the group, but grateful that you didn't try to talk me out of what I wanted to do." He became a good recruiter for me, too.

You do not need a big budget to develop people. Look for ways to develop technical or domain skills in daily work. Coach to increase organizational savvy and delegate to build management skills. It is one of the best ways to retain valuable employees -- for you and for your organization.

Conclusion

Great management is not easy, but it is within reach. Apply these practices consistently, and you will be well on your way to being a great manager. You will leverage other's work by focusing on the highest priority work and enabling productivity. You will help people grow and develop by helping them see their work in context, by providing course corrections, and by giving them opportunities to develop new capabilities. And (maybe best of all) you

will look good to your manager, too!◆

References

1. Derby, Esther, and Johanna Rothman. Behind Closed Doors: Secrets of Great Management. Raleigh, NC and Dallas, TX.: Pragmatic Bookshelf, 2005.
2. Shellenbarger, Sue. "Juggling Too Many Tasks Could Make You Stupid." CareerJournal <www.careerjournal.com/columnists/workfamily/20030228-workfamily.html>.
3. Weinberg, G.M. Quality Software Management: Vol. 1 System Thinking. New York. Dorset House, 1992.
4. Patrick, Francis S. "Program Management – Turning Many Projects Into Few Priorities With TOC." National Project Management Institute Symposium, Philadelphia, PA. Oct. 1999 <www.focusedperformance.com/articles/multipm.html>.
5. Davis, Noopur. "Self-Directed Teams: One Case Study." Pres. at Software Development Best Practices Conference and Expo, 17 Sept. 2003.

About the Author

Esther Derby is well known for her work in helping teams grow to new levels of productivity and coaching technical people who are making the transition to management. She is one of the founders of the Amplifying Your Effectiveness Conference and is co-author of *Behind Closed Doors: Secrets of Great Management*. Her latest book is *Agile Retrospectives: Making Good Teams Great*. Derby has a master's degree in organizational leadership and more than two decades experience in the wonderful world of software.

3620 11th AVE S
Minneapolis, MN 55407
Phone: (612) 724-8114
Fax: (612) 724-8115
E-mail: derby@estherderby.com