Readings For Problem-Solving Leadership

Second Edition

Esther Derby
Don Gray
Johanna Rothman
Gerald M. Weinberg

PSL Reader

Esther Derby, Don Gray, Johanna Rothman and Gerald M. Weinberg

This book is for sale at http://leanpub.com/pslreader

This version was published on 2020-03-31

Contents

Part 1: Leadership Styles

Managing in Mayberry

Do We Have to Choose Between Management and Leadership?

Beyond Blaming

Managing In Mayberry

Near the Blue Ridge Mountains in North Carolina, not far from where you think it should be, there really is a town called Mayberry.

Although the main highway bypassed the town years ago, the namesake for the popular 1960s television series is still a bustling community, and a fair amount of traffic enters Mayberry's downtown from the north on the US Highway 52 business spur every morning. In town for a week of consulting work, we were able to observe the recent road construction along that route and watched a trio of local citizens demonstrate their own unique management styles. Let's take a look at how these characters traffic management closely parallels common styles of *software project* management.

When road work just north of town closed Business 52, all the traffic entering town from the north had to take the 52 bypass around to the west side of town and enter the downtown on Key Street. Unfortunately, this meant traffic would have to make a left turn onto Key Street, crossing fairly busy east-west traffic (see Figure 1).

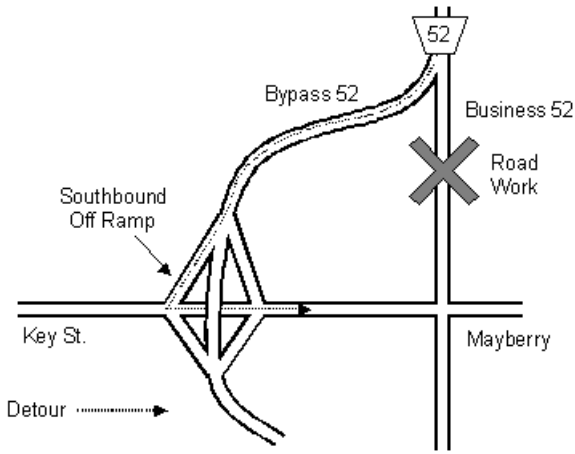

Figure 1

The town council feared that during the morning rush the traffic waiting to make the left turn onto Key Street would back up on the southbound off-ramp all the way to Highway 52 itself. This could cause a serious accident, since Highway 52 has a 65 mph speed limit. So the council decided to station one police officer and one or two rescue squad volunteers at the intersection to make sure that traffic on the ramp did not back up.

Three Approaches to Managing

Being a take-charge guy, the officer on duty (we'll call him Barney) arrived at the scene Monday and quickly sized up the situation. He decided that what was needed was a traffic light at the intersection of Key Street and the ramps. Since it would take the county months to approve a light, he decided to operate as a "human traffic light," directing traffic manually. Each direction got its turn: westbound Key (including left turns onto the southbound ramp), then eastbound Key (including right turns onto the southbound ramp), then the off-ramp (which could turn either way onto Key).

Barney's plan didn't actually work all that well. Traffic stalled in both directions on Key Street. And there were a couple of close calls on the ramp; traffic backed up almost onto Highway 52 once when Barney let a few cars turn left onto Key Street. By the end of rush hour he was hot, tired, and a little discouraged, and he had written a fistful of citations to drivers for making unmentionably rude gestures at a law enforcement officer.

On Tuesday, one of the rescue squad volunteers (a helpful local woman known as Aunt Bea) said she knew how to take care of the situation. She figured that traffic could probably take care of itself as long as drivers didn't have to cross each other's paths. So she let traffic go both ways on Key Street, and let people make right turns onto and off the ramps. When somebody had to turn left, she'd stop the other lanes and let them go. Aunt Bee's approach worked better than Barney's (at least nobody made rude gestures at her), but there was still a lot more congestion than we expected, and by the end of rush hour Bee was glowing profusely.

On Wednesday Sheriff Andy showed up, bringing a lawn chair and a thermos of lemonade. He set up the lawn chair on a shady spot from which he could see the intersection and a fair way down the off-ramp, and sat down to sip lemonade. When traffic started to back up on the ramp, he got up, stopped Key Street traffic, and let the ramp empty; then he went back to his lemonade. Other than that, Andy pretty much didn't seem to *do* anything. Despite his apparent inaction, the intersection just seemed to work. People were calm and relaxed, with the drivers making right turns creating breaks for others making left turns, and everything worked a lot like it did before anyone showed up to help—just a little better.

Putting on our consultant hats, we realized we'd just witnessed three distinct styles of management—Barney's *micro*management, Aunt Bee's *motherly* management, and Andy's *masterly* management. Since these styles are also common in software project management. Let's look at each of them in more detail, and see what we can apply to our own software projects.

A Question of Style

Each of our managers made different assumptions that shaped their style—in particular, assumptions about the *people* being managed, and about the *role* of the manager. These assumptions determined how they approached the critical activities of managing. In his book, *Quality Software Management, Vol. 1: Systems Thinking*, Jerry Weinberg highlights five critical activities:

1. understanding ***the problem to be solved***,
2. ***planning*** the solution approach,
3. ***observing*** what the people being managed are actually doing,
4. using ***rules and process models*** to determine what to do next, and
5. taking action*** to guide the group toward its goal.

Together these activities form a feedback system that "steers" the project team. How they are executed (i.e., what the manager defines as the problem, how the manager plans, what observations get made, which rules get followed, and how the corrective actions get taken) makes all the difference—determining just where the team will go, how the team members will feel about the software project as a whole, and ultimately how satisfactory the results will be.

Micromanagement

Barney practiced micromanagement, which is based on the assumption that the manager has to see to it that everything gets done. Most micromanagers don't deliberately meddle out of a need to be in control; they're just operating under the assumption that if they don't do it, it won't get done. Micromanagers also tend

to make the related assumption that those being managed will do what they're told to do; no more, no less.

These assumptions describe machines better than they do humans. Indeed, when Barney said we needed "human traffic lights," he was describing a situation in which both the manager and those being managed were more mechanical than human. Perhaps this is why so many good programmers become micromanagers when they get their first promotion—they're just "programming" the "bio-robots" who work for them!

Using Weinberg's model, we can see how Barney's assumptions defined his view of the critical management activities:

1. The ***problem to be solved*** was to personally make sure everything was done in an orderly fashion.
2. The ***plan*** that followed was for Barney to pretty much do everything himself. He would personally direct the movements of each and every vehicle. This meant that the plan had to be simple enough that he could be in control of its execution at all times.
3. Even with the simple plan, Barney was far too busy directing traffic to ***observe*** much. Standing in the middle of the intersection, he wasn't in the right position to see up the ramp when traffic began to back up onto Highway 52.
4. Even if he had made better observations, his manager-centered ***process model*** didn't allow him to do much. The underlying assumption that he was personally responsible for each and every car going through the intersection meant that he couldn't delegate much - he couldn't count on the drivers to do anything other than what he told them to do.
5. Barney's ***actions*** were pretty limited; because he had to control each vehicle, he couldn't leave his spot in the middle of the intersection. In the end, he couldn't do much beyond try harder at what he was already doing—waving his arms

> more frantically at the folks, in the hopes that they'd get through faster.

Because the manager must make (or at least approve of) all decisions, only one thing happens at a time and everything else lines up waiting for a turn. When simplicity, centralized information, and oversight are turned from virtues into vices, it creates a choke point that affects project planning and execution.

Simplicity Since the entire project plan must be under the control of the manager at all times, the plan must be simple enough that a single person can comprehend it in its entirety. This sets an upper bound on project complexity—if the problem to be solved is beyond this bound, the manager has to simplify it somehow (e.g., letting traffic go in only one direction at a time). This serialization of activities is a common simplification in micromanaged projects as well, and it wastes both effort and time. When serialization isn't enough, the manager may start leaving "non-essential" activities out of the project plan. Micromanagers are notorious for oversimplifying, to the point where their software project plans may leave out something essential for a successful product launch.

Centralized information Since the manager is the only one who can make a decision, it's critical that he get lots of quality information about how the project is doing. Unfortunately, the only observations allowed are those that the *manager* puts in the project plan—but that manager's far too busy making each and every decision to actually observe much of anything. So in practice, micromanagers are often flying blind, making decisions on little or no actual information.

Oversight The need to get explicit approval for each action adds to the amount of time required to accomplish tasks. So micromanagement tends to be inefficient, with a lot of people waiting around for the manager to tell them what to do next. The manager-as-bottleneck is a key structural problem. The practice also leads to people problems, such as initiative squelching. The manager's assumption

implies that the people being managed have nothing to contribute beyond the functions defined for them by the manager. What if the workers want to do something other than follow the rules—because they see a better way or a problem with the plan? Forget it. The micromanager will not allow it to happen. This creates short tempers and long days for those who are micromanaged.

Most people don't like this style of management. Some will respond with a sort of dead, mechanical compliance, waiting dutifully for their next set of instructions from the manager. Others may choose some form of subtle rebellion, such as "working to rule"—following the manager's instructions to the letter, no more, no less, even when those instructions are clearly a recipe for failure. And others will rebel more openly, taking advantage of the manager's continual distraction to get away with whatever they can. Alas, these responses to micromanagement tend to set up a positive feedback loop that reinforce the micromanager's assumptions and leads to even more micromanagement. Micromanagers tend to be very busy people.

So, is micromanagement ever appropriate? Certainly, when the problem to be solved is small enough for one manager to truly comprehend the entire project plan, and the people doing the work are willing to follow each and every command of the manager. While this situation can occur now and then, it's not very common in the software world.

A common cause of micromanagement is the newly promoted, technically competent manager rushing in to help a floundering employee or rescue a particular part of a software project. This creates a co-dependent dynamic where the manager becomes the rescuer, and the employee becomes helpless. This ensures that the next time there is a problem, the manager will step in again, and so on, until something happens to break the pattern.

While micromanaged projects can (and often do) result in successful product launches, it's often more in spite of their management

than because of it. There ought to be a more efficient and less aggravating way to handle the situation.

Motherly Management

Aunt Bea chose a kinder, gentler style that we call motherly managing, allowing the drivers to do some things for themselves, and helping them when she thought they needed help. But her underlying assumption was still pretty close to Barney's: the people being managed might be able to do a few routine things without being told, but all significant decisions—especially when there was some form of contention or competition—were still firmly under her control.

If the micromanager views the people being managed as machines, the motherly manager sees them more like children, able to do a few routine things but still needing protection from anything potentially dangerous. Like the micromanager, the motherly manager is not necessarily malicious or desperately in need of control. Aunt Bea had no great need to have power over the drivers; she just knew that they couldn't make major decisions without her help. She simply couldn't visualize the situation where one person could be turning left into the gap created by another turning right, because she couldn't see who was controlling the\ relationship, and she knew that two drivers certainly couldn't cooperate without somebody to coordinate them.

Aunt Bea's motherly assumptions defined her view of the key management activities:

1. The ***problem to be solved*** was something like "take care of the people who have to cross other traffic." Like Barney, she saw the problem in personal terms; it was her problem, not the drivers' problem.

2. Because Aunt Bea saw the drivers as human beings who could do a few things for themselves, her ***plan*** was a bit less rigid than Barney's. She could allow at least a few routine things to happen in parallel, but under exceptional conditions she would take full control of everything, which meant reverting to serial execution.
3. Aunt Bea's more distributed plan required somewhat more sophisticated ***observations*** than Barney's. She had to observe those situations in which her help was needed—in particular, left turns. Notice that she *wasn't* observing whether people were having trouble making left turns; her underlying assumption said that a left turn signal was a request for help. Like Barney, she spent her time in the middle of the intersection, a point from which she couldn't see up the ramp very well.
4. Because of her motherly assumption that the people being managed couldn't handle any form of contention or conflict, Aunt Bea's ***process models*** dictated that she must personally resolve these things. So her response to just about any out-of-the-ordinary condition was to stop traffic and go back to taking turns.
5. Like Barney, Aunt Bee was working from a very limited set of ***actions***, in part restricted by her need to be in the position of control at the center of the intersection. If those actions didn't work, about all she could do was more of what she was already doing.

Like micromanagement, motherly management can work when its underlying assumptions are true and the problem and solution aren't too complex. Trouble is, most software development shops aren't day care centers, and most development is non-routine and requires that a lot of conflicts be resolved. Interfaces, partitioning, decomposition, protocols—these are all "left turns" in the view of a motherly manager, who must personally make sure that everybody plays well together. This creates a structural problem similar to

micromanagement. Similar, but also different. Since some work can take place independently under motherly management, the manager is less of a choke point than in the case of micromanagement.

But because the process is still highly manager-centric, the actual amount of work that can be done in parallel is often less than expected. We end up with a process that's very nearly effective: almost parallel, relatively observant, and coming awfully close to giving workers independent responsibility:

Parallel (almost) Only pre-defined "routine" things can take place in parallel. As long as traffic went straight ahead or turned right, Aunt Bea's plan seemed to work. But she couldn't predict how many people would want to turn left. When lots of people started turning left, her plan fell apart. In the same way, the actual performance of a motherly-managed software project depends heavily on just how much of the development is really "routine" with no need for interactions or conflict resolution. If there are a lot more "exceptions" than expected, a lot of developers working in parallel according to the project plan may be sitting on their hands waiting for the manager to make a decision. This can make a project plan that was *parallel* in theory become *serial* in practice.

Myopic Motherly managers make more observations than micromanagers, but they still confine those observations to specific conditions noted in the project plan. If the conditions defined by the manager are in fact not the key exceptions that need to be managed, the motherly manager will be spending time and energy observing the wrong thing, while missing the observations that are really necessary for project success.

Nannying Motherly management can be less oppressive than micromanagement for the people being managed, because the "mother" allows her "children" to do a few things on their own. The individual developers can go ahead as long as they aren't going against the flow or getting into conflicts. But at the first indication that something non-standard is going on, the whole process stops

until the manager decides what to do. The manager must handle all the decisions that really matter—and this squelches the individual contribution to solving the overall problem just about as effectively as micromanagement. There is a great deal of variation here—a manager who views the employees as teenagers is less openly controlling than one who views them as toddlers. Still, most of the people who work in the software business have college degrees, and we wonder if we're making the best use of their expensive educations when we manage them as though they were children.

If we are going to find a style that's more efficient and effective than *micro* and *motherly*, we must start by changing our underlying assumptions. Barney sees the people being managed as machines to be programmed; Bea sees them as children to be helped. Now let's see what happens when Andy views them as adult human beings.

Masterly Management

Andy took an approach that at first didn't look like "management" at all. He just sat in his chair, sipping lemonade and watching traffic on the Highway 52 off-ramp. When it started backing up badly, he strolled out into the intersection, stopped traffic on Key Street, and let the off-ramp clear; then he went back to his lemonade. He seemed to be "working" a lot less than Barney or Aunt Bee, yet traffic flowed smoothly. We refer to Andy's style as *masterly*management — because of our three traffic controllers, only he was truly the master of the situation.

The keys to Andy's management style were his underlying assumptions: that drivers are adults, that most of the time they can take care of themselves, and that his role as a manager is to support these competent adults so they can do the real work of getting themselves safely through the intersection. This is vastly different from Barney's and Aunt Bea's assumption. Andy felt secure enough

about his own competence and the drivers' know-how that he could remove himself from the center of the job.

Because Andy did not place himself at the center of the management task, he could be much more flexible and effective at the key management activities:

1. Andy saw the ***problem to be solved*** as moving traffic efficiently and safely through the intersection. He also realized that most of the time this intersection didn't need any help; people made turns here every day without any supervision. What made this a unique problem that might require some management intervention? The detour increased traffic on the Highway 52 off-ramp, and that might, on occasion, cause traffic on the ramp to back up onto the highway and cause a safety hazard. Notice the difference—while Barney and Aunt Bea defined the problem in terms of what they had to do, Andy defined the problem in terms of results, independent of who actually "did the work." By doing this, Andy positioned himself to observe and "steer" the system that did work, rather than as the person doing the work.
2. With his understanding of the real problem to be solved, Andy was able to construct an effective ***plan*** for its solution. The drivers could be responsible for getting themselves through the intersection. He and his "management team" would monitor the off-ramp and make sure that it could be emptied when (and if) it backed up far enough to pose a safety hazard. While Barney might accuse Andy of not having much of a plan, the fact is that Andy's simple-looking plan actually allowed some very complex things to happen. Because he didn't attempt to control low-level actions by the drivers, Andy's plan delegated management work to individual drivers. This allowed them to operate in parallel, which they did—drivers waiting to turn left off the ramp took advantage of gaps in traffic created by drivers turning right.

3. Now that he had both a problem statement and a plan, Andy could identify which ***observations*** he needed to make. To keep traffic from backing up onto Highway 52, he had to watch the ramp—not the intersection. So he positioned himself off to the side, where he could see the ramp. This is another critical difference in Andy's style. Standing in the middle of the intersection, Barney and Aunt Bea were taking in a great deal of information—most of it irrelevant to solving the real problem. They weren't in the right place to make the observations that really matter. Of course, Andy didn't ignore what was happening in the intersection—but he didn't make the intersection his primary focus.
4. Andy's management style used two ***process models***. First, if traffic's backing up on the off-ramp, stop traffic on Key Street and allow the ramp to drain. Second, if something blocks the intersection, get it out of the way immediately. The rest of the time, Andy's process model says "let the drivers take care of themselves."

Both of these models are more subtle than they look. The first model allows Andy to do some fine-tuning as the morning progresses. How far up the ramp is "too far" for traffic to back up? At first he took a conservative approach, draining the ramp when it was backed up about halfway to the highway. Later, after observing how quickly Key Street traffic could be stopped to drain the ramp, he changed his definition of "too far" to something more like three-quarters of the way up the ramp. This meant even fewer interventions were needed, because often traffic would back up to the halfway point and then drain back down by itself.

The second model contains a flexible definition of just what triggers action. Andy's looking for a symptom, which could have a variety of root causes. If something blocks the intersection (e.g., a driver too timid to turn left), Andy's model will handle it.

1. Finally, Andy took a lot less "overt" ***action*** than either Barney

> or Aunt Bea. Most of the time it appeared that he was doing nothing at all. Yet, when action was required, he knew what action was appropriate and effective. But it would be wrong to say that Andy's actions were simpler than Barney's or Aunt Bea's. In fact, his infrequent interventions required *more* skill. After all, Barney and Aunt Bea were already standing in the middle of the intersection, and had the drivers' complete attention. Andy had to enter an intersection full of moving vehicles, get the drivers' attention, temporarily interrupt their self-management, get the drivers to carry out his instructions, and finally re-establish the self-managing system. This is a task requiring some skill.

Like the other two styles we've discussed, masterly management works when its underlying assumptions are valid. In software development, where the people being managed are skilled, competent, educated adults, these assumptions are usually *true*. Masterly management, therefore, addresses the structural and behavioral problems we saw with micro and motherly management:

- The delegation inherent in the plan means that most contentions and minor conflicts get solved without the manager's intervention, so most of the time the people aren't waiting for the manager's attention. When a problem does require the manager's attention, that problem doesn't have to wait in line behind a bunch of minor conflicts.
- This support for parallel activities means that masterly management can work with projects that are just too complicated to be understood in all their detail by a single manager—and most software projects would fall into that category.
- Because the people being managed are also delegated a self-management job, they are able to contribute observations that a micro or motherly manager is likely to miss.
- Masterly management involves managing the *project* rather than the *individuals*. Most of the time, the people doing the

> work are free to pick their own methods within some basic guidelines (for instance, driving on the correct side of the road, or using the corporate standard tool set). This allows creative energy that might otherwise be spent on finding ways to "beat the system" to instead go toward creating profitable products.

In short, a masterly manager like Andy observes and steers a system. If the problem is well understood, the plan is appropriate, and the people doing the work are competent, the controller often doesn't need to do much. Unlike micro and motherly managers, masterly managers spend most of their time in observation and thought rather than in frantic activity. But don't be fooled—when Andy was sitting in his chair sipping lemonade, he was more effectively in control of the situation than either Barney or Aunt Bea.

If masterly management is so good, why don't we see it more often? Because in some ways it's unsettling, especially for the manager:

Looks can be deceiving Masterly managed projects often give a certain appearance of chaos. When Andy managed the intersection, traffic was turning every which way, which was disturbing compared to the neat and orderly behavior when Barney was in charge. However, more traffic moved through the intersection, and did so more safely, under Andy's chaotic-looking management style. Many software projects already look like chaos. Will going to masterly management make them more so? We doubt it; we suspect that much of the apparent chaos in software development comes from resistance to micro and motherly management.

Power is as power does Masterly management requires a different mindset. Most people associate the word *manager* with the word *power*. Yet moving from micromanagement to masterly management involves giving up much of the apparent power and authority of the managerial position, and giving it to the people being managed. The masterly manager has more real power, according

to writer Barry Oshry (quoted in Weinberg's book *Becoming a Technical Leader*), if we define power as the ability "to act in ways which enhance the capacity of our systems to thrive and develop in their environment."

Measuring what counts In some organizations (particularly those where micromanagement is the rule), a masterly manager may have a hard time getting promoted. After all, you won't be doing much visible *managing* compared to the micro and motherly managers around you, and it will be easy for the micromanager who makes promotion decisions to conclude that the project succeeded in spite of your "inaction," not because of it.

But masterly management also has rewards. Masterly managers often don't have to work as frantically as micro and motherly managers. As a masterly manager, you're less likely to find yourself in the office at three in the morning, trying to resolve yet another trivial issue. And you'll get the satisfaction of knowing that you're truly an effective leader when the project team says, "We did this ourselves."

Micro, Motherly, or Masterly Management

The best way to determine your management style is to ask questions and observe what is happening.

- Do the people reporting to you scatter like leaves in the wind when you show up? Do you feel like they are performing to the letter of the law and not the spirit? Do you jump in and start coding when there is a problem? If so, you're probably micromanaging.
- Do you organize workflow for a minimum of interaction so things go smoothly in the team? Do you step in and try to

make everything all right for everybody? In crunch mode, do you revert to micromanaging? Your heart may be in the right place, but you may be in motherly managing mode.

- Do you spend a fair amount of time observing what is happening, thinking about the impact the events will have on your team and project, and planning what to do? If so, you may be masterly managing.

If you would like to change your management style, there are some important questions to think about. First, how did you come to have your current management style? For most of us, the way we manage is influenced by the people who've managed us, and by the environment in which we manage. Acknowledging these influences, and the constraints of your current work situation, may help you determine whether it's time for new models. It's important, too, to examine how you *feel* about your style. If you're happy with the status quo, change may not be necessary. But if you feel overworked, and seem to be constantly fighting fires, then maybe a change is in order.

And finally, what would you *like* to have happen? We saw that Barney, Bea, and Andy's view of the "problem at hand" shaped their unique responses, and the same is true for you. Once you know what you would like to have happen, you can create and implement the plans that will allow you to achieve your goals and keep your traffic running smoothly.

Do We Have to Choose Between Management and Leadership?

In a recent discussion on the state of a software company, a programmer declared, "We don't need managers around here, we need leaders!" I'm always puzzled by statements like this.

"How do you see the difference between management and leadership?" I asked.

"Managers do things right, and leaders do the right thing," the programmer replied, repeating a Warren Bennis quote.

"But what do they do differently?" I pressed.

"Managers manage, and leaders lead," the programmer replied with conviction.

Here's how leadership professor John Kotter describes the difference between management and leadership (which I paraphrase here):

Management is:

- establishing timetables and steps for achieving needed results and allocating resources to make it happen.
- creating structure, staffing and delegating responsibility, and having the authority to accomplish goals.
- monitoring results, identifying deviations, and planning and organizing to solve problems.
- producing key results expected by various stakeholders.

Leadership is:

- establishing direction, and developing a vision for the future.
- aligning people, modeling the vision, influencing, and creating teams and coalitions.
- inspiring people to overcome barriers to change by satisfying basic human needs.
- producing useful change.

Reading these lists, it's clear to me that organizations need both.

Here's an example. A test manager takes a job with a new testing group. He talks with his team, his manager, and the internal and external customers for his unit's work. Based on what he hears, he articulates a mission for the group: "We provide assessments of product quality and help product owners understand risks." That's leadership—setting a direction.

He works with the team to identify all the work they're currently doing, work that's in queue, and projects scheduled for the next several months. Together, they assess what they can accomplish, what they won't do, and whether they have the right mix of skills to do the work. That's management.

He supports the team as it self-organizes to accomplish the work. The organizing part is management (done by the team), while supporting self-organization is leadership—meeting human needs for autonomy.

The test manager works with the team to identify the resources they need—machines, tools, and training—and then adjusts the budget to acquire the necessary resources. That's management.

He's showing leadership when he meets with members of the team to understand their aspirations and help them articulate professional development goals. When they work together to build skills into daily work, that's management.

As the team works to test its products, the manager and the team work together to develop metrics and dash boards that show test progress and communicate the quality of the product—management again.

He makes sure the development manager and product owner define release criteria, leading through influence. He also brings change to the way the company makes ship decisions. When a testing project starts slipping, he pulls the team together to assess the issues and re-plan their approach—management, according to Kotter's definition.

And so it goes—a little management here, some leadership there. The balance shifts, depending on the situation. The test manager combines management and leadership activities to attend to people and accomplish meaningful work.

I've worked with people who were all leadership. When they lacked management behaviors—follow-through and attention to practical implementation—they left chaos in their wakes (and didn't actually produce much useful change). I've also worked with people who were mostly management, which only worked when they had enough personal warmth to navigate human relationships. (In accounting areas, you don't necessarily want creative ideas or big charisma—–think Enron.)

Viewing leadership and management as dichotomous sets up a false choice. Most positions in organizations need both, and that's what effective managers deliver.

Bibliography: Problem Solving Leadership

Adams, James L. 1974 *Conceptual Blockbusting.* San Francisco: W.H. Freeman. A classic work on problem solving that identifies some of the major blocks—intellectual, emotional, social, and cultural–that interfere with ideation and design.

Connirae Andreas with Tamara Andreas 1995 *Core Transformation, Reaching The Wellspring Within* Moab, UT. Real People Press. ISBN 0-911226-32-X The authors offer a gentle process that allows one to transform personal limitations and reach the wellspring within. The process can be useful whenever it is important to find out what you really values and why. Examples of such times might be negotiating anything, exploring resistance, or discovering hidden motivations.‘

Block, Peter 1987 *The Empowered Manager: Positive Political Skills at Work.* San Francisco: Jossey-Bass. Clearly and incisively written, this book makes a strong statement in support of participative, as opposed to “bureaucratic”, management styles. Block outlines a vision of how organizations can be and explains ways for people in management positions to make them so.

Bolton, Robert 1979 *People Skills:* How to Assert Yourself, Listen to Others, and Resolve Conflicts. Englewood Cliffs, NJ: Prentice-

Hall Other leadership qualities will be negated if you lack "people skills." Every person, no matter how skilled a leader, could probably benefit from this review of the fundamentals.

Carnegie, Dale 1936 *How to Win Friends and Influence People.* New York: Simon and Schuster. The principles of leadership haven't changed in 70 years, or 7,000 (though there is a revised, modernized edition of this self-help classic). If you find you cannot tolerate Carnegie's rather ordinary rules, you're probably not ready to be a leader among rather ordinary people.

Thomas F. Crum 1987 *The Magic of Conflict, Turning a Life of Work into a Work of Art.* New York. Simon & Schuster. ISBN 0-671-63818-1 The author shares his discoveries in the art of conflict resolution, gained in part, through his many years teaching Aikido, a Japanese martial art. His teaching goes beyond the combative forms to show how the power of harmony and love can work in even the most difficult of situations.

Esther Derby & Diana Larsen 2006 *Agile Retrospectives: Making Good Teams Great.* Pragmatic Bookshelf. Teams don't improve by accident. They improve when they reflect together on what they are doing, what's working, and what to do differently. This book tells you how to help a team inspect and adapt their methods and teamwork for continuous improvement.

Esther Derby Website[1] And her Blog[2] Esther's musings on the wonderful world of software and working with people in software.

Doyle, Michael, and David Straus 1976 *How to Make Meetings Work.* Chicago: Playboy Press Doyle and Straus have developed the "interaction method" for organizing and operating meetings of all types. Using this method, clearly described in this book, dozens of our clients have converted their meetings from the worst of times to the best of times. (For specialized technical meetings, see also Freedman and Weinberg.)

[1] http://www.estherderby.com
[2] http://www.estherderby.com/category/insights

Edwards, Betty 1979 *Drawing on the Right Side of the Brain.* Los Angeles: J.P. Tarcher, Inc. Edwards explains how she teaches drawing by using the brain model of hemispheric specialization. An excellent book for developing the generally under-used "creative" right hemisphere— for anybody, aspiring artist or not.

Freedman, Daniel P., and Gerald M. Weinberg 1982 *Handbook of Walkthroughs, Inspections, and Technical Reviews.* New York: Dorset House. Meetings are ritual forms of group interaction which can be designed for different problem solving purposes. This is a handbook on how to design and lead meetings for one type of problem solving - obtaining critical reviews of work in progress. Such reviews can be a source of growth, or great anxiety and conflict, depending on how they are designed and led. (For some general principles of meeting design, see Doyle and Straus.)

Gause, Donald C. and Gerald M. Weinberg 1982 *Are Your Lights On?* New York: Dorset House. In this book, Don and Jerry have tried to teach people some things about the art of problem definition. It's light enough to be a catalyst between systems people and normal human beings. The process is continued in a more formal way in:

1989 *Exploring Requirements: Quality Before Design.* New York: Dorset House. A survey of human processes that can be used in gathering complete, correct, and communicable requirements for a software system, or any other kind of product.

Gordon, Thomas 1977 *Leader Effectiveness Training: The No-Lose Way to Release the Productive Potential of People.* New York: Wyden Books. Gordon, a management consultant, is also the author of the tremendously popular and helpful Parent Effectiveness Training. Many potential problem solving leaders will find useful material in Gordon's "no-lose" approach.

Don Gray's site and blog[3] Integrating People, Projects, Processes

Gross, Ronald 1979 *The Lifelong Learner* New York: Simon &

[3] http://www.donaldegray.com/

Schuster An essential handbook for those who take responsibility for their own learning. Full of ideas, suggestions, and specific resources for self-renewal.

Hawken, Paul 1987 *Growing A Business* New York: Simon & Schuster. This delightful and very practical book is by the founder of Smith & Hawken, a mail-order garden tool company known for its integrity, product quality, employee satisfaction, and strong customer focus – all of which the author believes are essential for a successful business. Although the book is written for the small-business person, its lessons apply equally well to companies of any size.

Hollander, Edwin P. 1978 *Leadership Dynamics* New York: Free Press. An entry point into the theoretical and experimental results about leadership. The references are thorough, but inconspicuous, so that you may start with ideas and move to sources if you wish.

Keirsey, David and Marilyn Bates 1978 *Please Understand Me: An Essay on Temperament Styles.* Del Mar, California: Prometheus Nemesis Books. This book gives a detailed and illuminating explanation of the Jungian model of "temperaments." It includes a self-administered test and scoring system–a simplified version of the well-known Myers-Briggs Type Indicator. It makes a strong argument for respecting human diversity in personality styles.

Kennedy, Eugene 1980 *On Becoming a Counselor.* New York: Continuum Publishing Co. Leaders often find themselves in the role of counselors to people asking for help. Kennedy's book is directed to those who are not professional counselors, but who often perform this function and want to know at least how to avoid doing harm.

Kroeger, Otto, and Janet M. Thuesen 1992 *Type Talk at Work.* New York: Delacorte Press. This very readable book reviews the Myers-Briggs model and applies it to the workplace, giving many useful and humorous examples.

Lynch, James J. 1985 *The Language of the Heart: The Human Body in Dialogue.* New York: Basic Books. An important and very

readable book about the interrelatedness of language, emotion, and health. Lynch, a psychologist, makes a convincing case for paying attention to the "social membrane" (human interaction) in understanding and treating hypertension and migraine headaches.

McKim, Robert H. 1980 *Experiences in Visual Thinking. 2nd ed.* Monterey, California: Brooks/Cole. This book focuses on visual and other non-cognitive approaches to creative thinking and problem solving. It includes an excellent series of exercises for the reader to practice.

Myers, Isabel Briggs 1980 *Gifts Differing.* Consulting Psychologists Press, Inc., 577 College Ave., Palo Alto, CA 94306 Myers, one of the co-designers of the famous Myers-Briggs Type Indicator, explains the Jungian theory behind it. She discusses how to interpret and make practical use of the revealed types, in This book could be read profitably in conjunction with Keirsey and Bates, Please Understand Me.

NTL Institute *Reading Book for Human Relations Training*, edited by Larry Porter P.O. Box 9155, Rosslyn Station Arlington, VA 22209 Many of our workshop graduates want to learn more about how they interact with other people. We often recommend that they take NTL's "Human Interaction Laboratory"—and this is the book of readings that they take home from that lab.

Oshry, Barry 1995 *Seeing Systems, Unlocking the Mysteries of Organizational Life.* San Francisco, CA. Berrett-Koehler Publishers. ISBN 1-881052-73-7 Oshry suggests that we need to understand the source of a problem, the kind of organization involved, and the implications of our structural position in that organization. The book is clearly written, thought provoking, and designed for immediate application.

Progoff, Ira 1975 *At a Journal Workshop.* New York: Dialogue House Library If you want to learn more about keeping a journal, here's a whole book on the subject.

Quenk, Naomi 1993 *Beside Ourselves: Our Hidden Personality in*

Everyday Life. Palo Alto, CA. CPP Books (A Division of Consulting Psychologists Press, Inc.) An excellent study of the MBTI "Fourth Function"—the state in which we are not ourselves but operating under stress.

Rogers, Carl 1977 *On Personal Power.* New York: Dell. If you are interested in power and leadership, read this book before you do anything else. Other books by Carl Rogers which will help you on your path to effective leadership include:

1961 *On Becoming a Person.* Boston: Houghton Mifflin

1980 *A Way of Being.* Boston: Houghton Mifflin

Rothman, Johanna. 2012 *Hiring Geeks That Fit.* Rothman Consulting/Leanpub. Your leadership skills are most evident to candidates during the hiring process. Johanna explains how to make the hiring process one that invites people in, not scares them off.

Rothman, Johanna and Esther Derby. 2005 *Behind Closed Doors: Secrets of Great Management.* Pragmatic Bookshelf, Raleigh NC and Dallas TX, 2005. You don't have to choose between management and leadership, but you do need to master the tactical parts of management. Johanna and Esther show you how.

Rothman, Johanna. 2007 *Manage It! Your Guide to Modern, Pragmatic Project Management.* Pragmatic Bookshelf, Raleigh NC and Dallas TX, 2007. Part of leadership is recognizing the context in which you work, and then managing to make that context as good as you can make it. Johanna helps you recognize your context, and helps you choose what to do next, so your projects can be successful.

Rothman, Johanna. 2009 *Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects.* Pragmatic Bookshelf, Raleigh NC and Dallas TX, 2009. Too many projects? Want to organize them and evaluate them without getting buried under a mountain of statistics? This book will help you collect all your work, decide which projects you should do first, second—and never.

Rothman, Johanna 2013 *Manage Your Job Search.* Rothman Consulting/Leanpub. Treat your job search like a project, dividing it into small steps and obtaining feedback along the way. Discover your traps. Learn from the tips. Best of all, find fulfilling work.

Johanna Rothman site and multiple blogs[4] Start here for Johanna's articles and blogs.

Russell, Bertrand 1951 *The Conquest of Happiness.* New York: Signet Books Unhappy people are not leaders. Nobel Prize winning philosopher Bertrand Russell tackles the ancient question of how to be happy—and succeeds. This classic is available free on line.

Satir, Virginia, John Banmen, Jane Gerber, Maria Gomori 1991 *The Satir Model, Family Therapy and Beyond.* Science and Behavior Books, INC., Palo Alto, CA 1991 ISBN 8314-0078-1

Satir, Virginia 1983 *Conjoint Family Therapy. 3rd ed.*, Palo Alto, Ca.: Science and Behavior Books.

1972 *Peoplemaking.*, Palo Alto, Ca: Science and Behavior Books.

1985 *Meditations and Inspirations.*, Millbrae, Ca.: Celestial Arts.

1978 *Your Many Faces.*, Millbrae, Ca.: Celestial Arts.

1976 *Making Contact.*, Millbrae, Ca.: Celestial Arts.

1976 *Self-Esteem.*, Millbrae, Ca.: Celestial Arts.

We have been greatly influenced by the work of Virginia Satir. We first became aware of her radical approach to life through Peoplemaking, which is a good survey of her ideas about how we learn to interact with others. Conjoint Family Therapy is more of a comprehensive textbook for therapists, but like all her books, it is written without academic pretension. For a lighter introduction to specific topics of importance to leaders, try one or all of her little books from Celestial Arts.

[4] http://www.jrothman.com

Waddington, C.H. 1977 *Tools for Thought.* New York: Basic Books. This posthumous work by the brilliant geneticist and teacher introduces ordinary people to sophisticated analytic methods that may be used in problem solving - for example, general systems theory and game theory.

Weinberg, Gerald M. 1986 *Becoming a Technical Leader.* New York: Dorset House. This book is a personalized guide to developing the qualities that make a successful leader. It identifies which leadership skills are most effective in a technical environment and explains why technical people have characteristic trouble in making the transition to a leadership role.

1985 *The Secrets of Consulting.* New York, Dorset House Publishing, Inc.

2001 *More Secrets of Consulting: The Consultant's Self-Esteem Tool Kit.* New York, Dorset House Publishing, Inc. Consulting may be defined as the art of influencing people at their request. *The Secrets of Consulting* takes you behind the scenes of that art, explaining in detail why the world of consulting seems so irrational, and some very practical steps you can take to make it more rational. *More Secrets of Consulting* offers a set of tools to help the consultant operate from a congruent position in all circumstances.

1991 *Quality Software Management, Volume 1: Systems Thinking.* New York: Dorset House.

1992 *Quality Software Management, Volume 2: First-Order Measurement.* New York: Dorset House.

1994 *Quality Software Management, Volume 3: Congruent Action.* New York: Dorset House.

1996 *Quality Software Management, Volume 4: Anticipating Change.* New York: Dorset House. These books apply our teaching about system thinking, observation, and congruent action to the context of software engineering. Volume 1 is based on the type of thinking that Satir applied to families. Volume 2 is structured around Satir's

Interaction Model. Volume 3 is based on Satir's ideas on congruence. Volume 4 is based on her Change Model.

1975 *An Introduction to General Systems Thinking.* New York: John Wiley & Sons.

Weinberg, Gerald M. and Daniela Weinberg 1979 *On the Design of Stable Systems.* [Retitled: *General Principles of System Design.*] New York: Dorset House. These two volumes form a good starting place for those whose interest has been aroused about the subject of general systems thinking. The first volume emphasizes the role of perception in thinking—"it's not the event, but the reaction to the event." The second volume emphasizes the systems principles that can be derived from the elementary need of any system to survive—illuminating the importance of Satir's "survival rules" in the most general possible context.

Weinberg, Gerald M. 2006 *Weinberg on Writing: The Fieldstone Method.* New York: Dorset House Publishing. Writing is an essential leadership skill. This book can help: A reviewer said: "A few years ago I went to the writing section of my local library and checked out every book I could find. I found better books about the business of writing, pitching stories, and fundamental English rules. As for the actual process of writing, this book is far and away better than anything else I have ever read.

Weinberg, Gerald M. 2007 *The Aremac Project.* New York: Little West Press, 2007. This is a story of several young problem-solving leaders and how they develop while coping with some thrilling adventures. A review saidL "A gripping story ripped from tomorrow's headlines. A great cast of characters, not one of them afraid to be smart … and with 'tude to spare. A fun read as a familiar name in books takes his first swing at the novel … and scores a home run!"—Mike Shepherd, national best selling author of Kris Longknife—Defiant. [Note: Each of Jerry's subsequent novels also concerns the adventures of problem solving leaders. They are all available as ebooks, as are all his non-fiction books.]

Weinberg, Gerald M. Website[5] *The Secrets of Writing and Consulting Blog.* http://secretsofconsulting.blogspot.com/ For consultants who want to share their secrets, plus occasional thoughts to help both the developing and experienced writer.

[5] http://www.geraldmweinberg.com

More From Esther Derby

I started my career as a programmer, and over the years I've worn many hats, manager, internal consultant and business owner. From all these perspectives, one thing became clear: our level of individual, team and company success was deeply impacted by our work environment and organizational dynamics. As a result, I have spent the last twenty-five years helping companies design their environment, culture, and human dynamics for optimum success.

How your company's values are reflected in your environment, your culture and organizational dynamics, directly determines the quality and level of your success. When your company environment, culture and organizational dynamics are positive, mutually receptive and reinforcing, your people and teams have the capacity to achieve great things.

I help teams and managers understand what's working and where there are contradictions that sap productivity and stifle innovation. We explore how best to maximize your capacity for achievement by eliminating wasted effort, politics, cynicism, and fear. Together, we achieve a holistic view of your organization, and design your environment to directly enable, support and sustain your agile success, now and into the future. *** I've written over 100 articles, and co-authored two books–Agile Retrospectives: Making Good Teams Great and Behind Closed Doors: Secrets of Great Management. I write about management, leadership, collaboration, organizations and change (or another topic I'm currently exploring). If you'd like to get a taste of how I approach things, many of my articles are posted on this site. If you have particular topic or issue you'd like to explore, email me, and I'll put together a collection of my articles that may be helpful to you.

I also teach workshops and talk to groups all over the world.

I've learned a lot about how organizations work through observation and action research. I also hold an MA in Organizational Leadership and a certificate in Human System Dynamics.

See more at: http://www.estherderby.com

More From Don Gray

I started working with clients in 1984 focused on delivering value. My experience crosses a variety of industries, from finance to manufacturing. My clients' sizes range from small startups to Fortune 50 organizations. This background allows me to assist clients as they change their software development practices.

Change contains many dimensions. Often my clients initially focus on a single aspect of their change, for example moving to the Scrum development framework. I help them succeed by ensuring individuals, teams and the company have the necessary ability and motivation to succeed.

Building on my control systems background, I've studied complexity theory, modeling, and network analysis. I balance this with understanding communication, different personality models, and human systems dynamics. I incorporate these diverse subjects into my work and writing. You can find a number of my articles published at Better Software magazine and StickyMinds.com.

Along with Esther, Jerry and Johanna, I helped create and host the AYE Conference and the Change Artistry workshop. I employ the same experiential construction for the public and custom workshops I facilitate.

There are a number articles[6] at my website and on my blog[7] about change[8], problem solving[9], systems thinking[10], and teams[11].

If you have any questions at all about change, teams, leadership, or

[6] http://www.donaldegray.com/article/article/
[7] http://www.donaldegray.com/category/blog/
[8] http://www.donaldegray.com/tag/change/
[9] http://www.donaldegray.com/tag/problem-solving/
[10] http://www.donaldegray.com/tag/systems-thinking/
[11] http://www.donaldegray.com/tag/team/

just want some feedback or a second opinion, feel free to email me.
don@donaldegray.com

More From Johanna Rothman

I consult, speak, and train about all aspects of managing product development. I have a distinctly agile bent. I'm more interested in helping you become more effective than I am in sticking with some specific approach. There's a reason my newsletter is called the "Pragmatic Manager"—that's because I am!

Want to read the other things I've written? Take a look at my articles[12] and blogs:

Johanna Rothman's Website[13]

Managing Product Development Blog[14]: Management, especially good management, is hard to do. This blog is for people who want to think about how they manage people, projects, and risk.

Hiring Technical People Blog[15]: Hiring technical people and being hired isn't necessarily easy, no matter what the economy is doing. Use the tips here to hire better, or find a new job.

Create an Adaptable Life Blog[16]: The only people who don't have to change are the ones who are already buried. Since you're not one of those, see what I've learned in living with and adapting to change.

If you liked this book, you might like the other books I've written:

- Agile and Lean Program Management:Collaborating Across

[12] http://www.jrothman.com/articles/
[13] http://www.jrothman.com
[14] http://www.jrothman.com/blog/mpd
[15] http://www.jrothman.com/blog/htp
[16] http://www.createadaptablelife.com/

the Organization[17]
- Manage Your Job Search[18]
- Hiring Geeks That Fit[19]
- Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects[20]
- Manage It! Your Guide to Modern, Pragmatic Project Management[21]
- Behind Closed Doors: Secrets of Great Management[22]

I'd like to stay in touch with you. If you don't already subscribe, please sign up for my email newsletter, the Pragmatic Manager[23], on my web site. Please do invite me to connect with you on LinkedIn[24], and follow me on Twitter, @johannarothman.

[17] https://leanpub.com/agileprogrammanagement
[18] https://leanpub.com/manageyourjobsearch
[19] https://leanpub.com/hiringgeeks
[20] http://pragprog.com/book/jrport/manage-your-project-portfolio
[21] http://pragprog.com/book/jrpm/manage-it
[22] http://pragprog.com/book/rdbcd/behind-closed-doors
[23] http://www.jrothman.com/pragmaticmanager/
[24] http://www.linkedin.com/in/johannarothman

More from Gerald M. Weinberg

I was very small when I was born–only 9 pounds or so–and I've never gotten over it. Inside me, there's a little boy who can't make sense out of the world, but keeps trying. I read about computers when I was about 11, and thought that these "giant brains" might help, so I determined to work with computers when I grew up. This led me to study math and physics, because my guidance counselor told me that computers "had something to do with electronics." There were no computer classes, or even computers, any place where I found myself. Indeed, I never took a computer course in my life. Yet.

After more than 50 years working with computers, I've learned a couple of things, but I still can't make sense out of most of it. Most of all, I've discovered that people are at the bottom of just about every problem–but I think I knew that when I was little, then got talked out of it somewhere along the way. I've worked hard at relearning this lesson, and learning how to do something about it. While educating myself, I learned a second principle: I'm the "people" at the bottom of most of my problems.

Anyway, using these insights, I've been able to help a lot of people solve problems. I've written a lot of books, too, which many people have told me are helpful; but I get much more satisfaction from helping people directly, so I get to know them. For instance, I've helped several hundred people write their books, and several thousand find better ways to do their job. I feel very good when I find out that many of these people have learned things from me that they take back with benefit to their family life–possibly because my early family life wasn't very wholesome.

From my first attempt to create a better family life for myself, I have four children (one, now deceased) and four grandchildren. Although I thought I had screwed up that family life, too, now that the kids have all reached the half-century, things seem to have worked out okay. I did learn a lot from that first attempt, which enabled me to choose Dani for my second—by far the best decision I ever made. We've now passed almost five decades of living together, which involved about 7 serious renegotiations of our original, informal marriage contract. We've also had 16 children so far, but they've all been German Shepherd Dogs.

I like to start new things that help people. Although I've written several hundred articles and more than 100 books, my greatest satisfaction is creating real-life learning experiences—schools, camps, institutes, seminars, or development groups. I guess one of the things I am is a social architect. When I'm not starting a new social system, I'm often repairing an old one; in this role, I'm an organizational therapist. And, I also like to work with one-person organizations; perhaps I am at my best as a restorer of works of art.

Anyway, if you're a work of art that needs a little restoration, I invite you to explore these readings and see if there's anything we can do for each other. You might also discover more about my work by exploring my website, http://www.geraldmweinberg.com. Or, you can examine the essays in *The Gift of Time*, edited by Fiona Charles, celebrating my 75th birthday with some truly fine essays from many of my students and colleagues.