

Practical Ways to

MANAGE YOURSELF

MODERN MANAGEMENT MADE EASY: BOOK 1



*Author of Manage Your Project Portfolio:
Increase Your Capacity and Finish More Projects*

JOHANNA ROTHMAN

Practical Ways to Manage Yourself

Modern Management Made Easy, Book
1

Johanna Rothman

ISBN 978-1-943487-12-7



Practical ink

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the author.

Every precaution was taken in the preparation of this book. However, the author and publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information contained in this book.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Practical Ink was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals.

© 2020 Johanna Rothman

7. When Do You Ask, “Are You Done Yet?”

Back when I was a Director of many things at one company, we had an urgent patch to go to a customer. My VP wanted it “yesterday.” Well, time only goes in one direction.

I gathered my Continuing Engineering team and explained the pickle we were in. “Everyone wants this patch right away. The customer is truly pissed. I want to know that we have a fix that works. And while you are working on it, I will need to know updates every morning and every afternoon. I will run interference for you as well as I can.”

Everyone groaned. They knew what this meant. We had a small company. The corporate management was just down the hall from our offices. Even though I said I would run interference, nothing would prevent the VP of Engineering, the CEO, or the CTO from popping their heads in “to see what’s going on.” Everyone wanted to make the customer happy, *right now*.

At the time, I didn’t know about kanban boards. I knew about spreadsheets and email. We had four full-time people working on this fix, a significant investment of people and time. Because they swarmed and left notes, I also knew what they planned and what they discovered.

They managed themselves. Their offices were close to each other. Every day, about noon or so, they gathered in my office, so I would have the most up-to-date status. It wasn’t quite a standup, because some of the work was what we would now call “spikes,” short timeboxed experiments. (At first, we had no idea what was causing the problem.)

As we identified the problem, I explained to management *on behalf of the team* how they narrowed down the problem and identified it. Then I explained to management *on behalf of the team* how they were debugging the problem. Then I explained to management *on behalf of the team* how they were testing the fixes they proposed. Then I explained to management how they were packaging the fix they had decided on.

If we'd had a visual board, this might have been easier. I used email. It took close to a month. It was a very difficult fix.

Notice what I did:

- I explained to the team the results I wanted: as quickly as possible, but it had to be right. Right trumped shoddy.
- I explained that I needed information and how often I needed it.
- I ran interference and kept the rest of the management team informed on a daily basis. My goal was no surprises.
- I explained things on behalf of the team so they got the credit. I was doing my management job, not technical work.

We had sufficient transparency with our management. Because they knew where we stood, I could share the interim results with the customer. The customer was not happy during this month, but they were pleased to know we were working on the fix. By the time they got the patch, they were very pleased. It worked.

I did not micromanage these people. I understood their state. There is a big difference.

If I had stood over their shoulders and asked, "Is it done yet?" I suspect I would have had different results.

My team understood that I was doing my management job. I wasn't able to prevent all senior management interference. But I prevented

most of it. In return, the team was free to work together to accomplish their goal: a fix that didn’t upset the rest of the system and solved this customer’s problem.

It’s easy to fall into micromanagement. All of us knowledge workers—managers and technical people—excel as problem-solvers. We want to help other people solve their problems.

However, problem-solving as micromanagement inflicts help on other people. The constant direction or status request are not helpful at all. Micromanagement irritates other people, and it prevents them from doing their jobs.

7.1 Myth: It’s Fine if I Micromanage

Sharon poked her head into Heath’s cubicle. “Hey, Heath, are you done yet with that fix?”

Heath turned around. “Sharon, you asked me that less than an hour ago. I’m not done yet.”

“Well, I need to know when you will be done. Oh, and I need to know if you’re using the design we discussed.”

Heath started to turn red. “*We* didn’t discuss any design at all. *You* told me a design to use. Because *you* used that design back in the day, back when you were a developer. So *you* want me to use it now. Are you delegating this fix to me or not? Do *you* want to do it?”

Damon popped up from his cube on the other side of Heath. He tapped Sharon on the shoulder before she could reply. “Sharon, it sounds as if you need information. It also sounds as if Heath needs time to finish that fix. How about I help?”

Sharon looked relieved. So did Heath. “That would be great,” she replied. “I have another Ops meeting in fifteen minutes where everyone is going to ask me when the fix will be done. I’d really

like to know the answer." She took off down the hall, texting on her phone as she went.

Damon sat down next to Heath. "OK, tell me what's going on. You sound as if you're at the end of your rope."

"I know this is a critical fix. But Sharon won't let me do my job," Heath said. "It's not just this fix; it's anything. She wants to design this fix for me. She's come over here five times this morning, and it's not even noon. OK, she has that meeting, and I know they're going to rake her over the coals. I got that part. But interrupting me isn't going to help. I need time to think."

Damon prodded. "You said it's not just this fix? What do you mean?"

"I've taken over Sharon's subject matter expertise areas, right? I want to share them. I don't want to be the only person who knows them. That's crazy. Every time I ask for help she says things like, 'I didn't need help when I was the developer,' or crazy things like that. This is a big fix. I said I want to pair with another developer or a developer and a tester, too. She told me I didn't need to. How could she possibly know what I need to do? Is she a mind reader now?"

Damon nodded.

"And for my regular work, when we estimate as a team, she's there," Heath continued. "She's not supposed to be there, but our supposed Scrum Master won't kick her out of the room, so she screws up our estimates. You haven't seen this, because you're a tech lead on another team. She tells us our estimates are too big, so she bullies us into making them smaller. But we can't make them smaller. They are what they are. It's crazy."

Damon sighed and nodded.

"Then she tells us how to design. She doesn't know the code anymore. She's in meetings all the time. But she thinks she does. This

is really bad. I don’t think I can take it anymore. Maybe I’ll quit today. That would serve her right.”

“Whoa,” Damon said. “I can tell you’re really frustrated.”

Heath sighed and nodded.

“I learned something important last year,” Damon said. “Managers are people, too. I suspect Sharon doesn’t know what to do in her new role. She hasn’t learned how to delegate or what an agile manager does, so she’s insinuating herself into the team. Has anyone provided her with feedback?”

“No. We’re just putting up with things.”

“OK. You folks have to learn how to give Sharon feedback. For now, I’ll talk to her. Would you like me to work with you so you have someone to talk with, just on this fix?”

“Yes, please,” Heath replied. “This is harder than it looks.”

“OK. I’ll timebox our work to ten minutes so I have something to say to Sharon. I’ll run interference for you and give her a status. Then I’ll come back to you, OK?”

7.2 Learn How to Delegate

If you were one of the best technical people and you were promoted to a manager, you may have to learn how to delegate. (See [Consider What You Can Delegate](#).) If people outside the team clamor for information and you’re not sure what to tell them, you might feel the urge to push the current technical staff aside and do it yourself. You might be faster. You might be right. But unless the technical staff ask you for help, that’s wrong.

People want to feel accountable for their own work. People need to both succeed and fail on their own. They are adults—treat them that way.

If you explain to people the results you want and the boundaries of what the acceptable deliverables are, people will deliver. And, as in this case, if you explain that you need information in a timely manner, they will deliver that, too.

7.3 Clarify Which Information You Need When

Sometimes managers micromanage when they need information. In that case, it's easier to create an information radiator (some sort of easy-to-access progress indicator) rather than have the manager come running to you every thirty minutes. Or you can work with a buddy so that someone else runs interference for you. This way, you can concentrate on your technical work, and some other manager will receive the information.

Often, a senior manager needs the information. You can ask your immediate manager to provide the cover for you. If that doesn't work, see if a tech lead or someone else who has the manager's respect will work with you. It's worth a shot.

7.4 Offer Feedback to Managers

Managers need feedback to know that they are micromanaging. They might not need to know when they are headed to the Ops meeting, but they need to know.

Damon spent those ten minutes with Heath and then went to debrief Sharon.

Damon said, “Here's where Heath is right now. I'll be working with him for the next hour, so you can be sure we will be making progress. And there's something else I want to discuss with you. Check with me when you return, OK?”

"No problem. Maybe by then, you two will have fixed the problem," Sharon replied.

When Sharon returned from the Ops meeting, she checked with Damon. "OK, I'm ready. Did you two fix the problem?"

"Not yet. Heath has a good handle on it right now. I'm going to work with him later. But I need to talk to you about something else."

"Oh, what's that?" Sharon asked.

"When you ask Heath for status that often and tell him how to design and implement, you're micromanaging him. Are you aware of that?"

"Well, no. I thought I was being a good mentor or coach. I thought that's what good managers did," Sharon said.

"No, good managers offer suggestions—if people want them," Damon said. "You can ask, 'Would you like help?' And if people say, 'No, thanks,' you back off. Believe me, I know how tough this is to take. Even as a tech lead, I want to tell people what to do sometimes. But I can't. I can offer, but I can't make them do things."

Sharon sighed.

"When you were promoted, did anyone ever tell you about delegation?" Damon asked.

"No, no one ever did," Sharon admitted.

"Hmm. Do you ever have one-on-ones with your manager?" Damon asked.

"Oh, no. Steve says he's too busy. I'm winging it," Sharon said.

"That's a problem," Damon said. "If you want, I can tell you what I know. My boss, Joakim, is a great manager. I'm just a tech lead, so I don't do 'management' per se, but I'm in a lot of the same quandaries. If you want, we can meet once a week and I can tell you what I know."

"That sounds great," Sharon said. "Thanks."

7.5 Recognize Your Micromanagement

Sharon thought she could still “do it all.” No one can. It’s a bad idea and incongruent if you try.

Ask yourself these questions to see if you’re micromanaging:

- How many times a day do you check in with people or a team?
How many times a week?
- How often do you ask people if they want your opinion or your help?
- How often do you answer for other people?
- How often do you tell people which risks you want them to manage?

Sometimes you answer on behalf of the team for what feels like a good reason. Your managers might want an estimate *now*. It’s very tempting to offer an estimate and placate your managers. Don’t do it.

Instead, practice saying something like, “I can ask the team for an estimate today, and I’m sure we can give you one within a couple of days. If that’s not fast enough, I will tell them to stop working on whatever they’re doing to create the estimate for you.” Your managers don’t need to micromanage you, either.

Sometimes you have experience with the design or architecture, and you’re worried about where the team is headed. If so, explain that you’re worried. Explain your concerns: “I’m worried the design won’t take into account these potential problems,” and list those problems. Then you can add, “Is there a time you can explain to me how you’re addressing those questions?” Wait for the team to explain. Extend the trust the team deserves.