

# Practical Ways to Lead an INNOVATIVE ORGANIZATION

MODERN MANAGEMENT MADE EASY: BOOK 3



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

**JOHANNA  
ROTHMAN**

# Practical Ways to Lead an Innovative Organization

Modern Management Made Easy, Book  
3

Johanna Rothman

ISBN 978-1-943487-18-9



## 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

# 13. Who Has the Power to Decide?

How many policies, procedures, and standards does your company have? Think about all these examples of decisions codified as policies, procedures, and standards:

- Accounts Payable for travel expenses.
- Administration for allowable furniture in the office—including the number and size of monitors.
- HR for when and how you can promote or reward people.

I've seen more:

- When managers sign off on work they don't participate in. You might have seen a "Change Control Board" composed of managers, not the peers of the people who do the work.
- When managers decide on core hours—not the team or the workgroup.
- When managers impose an "estimate" on a team and expect the team to deliver to that date.
- When managers decide on any process they don't use themselves. For example, when managers decide which approach or framework a team will use.

When I ask people if there's a name for these policies, procedures, and standards, they almost always say, "Bureaucracy!"

Many of these policies arose because we want to prevent Something Bad from happening. Or, someone did Something Bad long ago.

That person might have paid for the infraction. The policies, procedures, and standards still punish the people in the organization.

These policies, procedures, and standards reinforce the status quo and can make innovation quite difficult. In addition, the policies, procedures, and standards reinforce hierarchical and centralized decision-making.

And, while managers can try to enforce a hierarchical and centralized approach to decisions, people talk to each other. Any given team or workgroup understands how they work and how they finish work.

And, if the people work on an overarching goal? They talk to each other as in [Figure 4: Reality of Information Across the Organization](#).

Why do managers make decisions for other people? Often out of fear, such as the risk that someone will make a mistake and that mistake will have significant consequences.

The people with the organizational power make those decisions. And, too often in my experience, the people who decide do not have to live with their decisions. Here are some examples:

- An international company had policies that anyone VP level or above could fly business class for international flights over five hours. The technical staff had to fly coach and did not have an opportunity to arrive a day early to catch up on sleep. At least 80% of the time, the technical staff had to discuss and decide on the product issues—not the managers.
- A company that wanted to manage their real estate footprint decided only managers would have offices—even though the managers rarely used their offices because they traveled so much.
- A company often advanced loans and other low-cost money perks to their managers. The technical staff waited weeks for travel reimbursement.

The result? The managers had plenty of autonomy. The technical staff had very little autonomy.

When managers make decisions like these, managers reinforce their power. Power is not good or bad—it is a fact. The question is, who uses which power?

I see many problems with policies, procedures, and standards:

- The more policies and procedures your organization has, the less experimentation and change you can encourage. The fewer experiments you can tolerate.
- The policy is often outdated and does not achieve its desired result.
- We rarely examine and remove outdated policies.
- Too often, the people who make the decisions do not have to live with the result of those decisions.

The policies, procedures, and standards reinforce management power. When managers don't relinquish power, people have to ask the manager for permission. That increases cycle time and too often creates underprivileged people and teams.

Management power tends to increase management control.

With decision power, managers tend to create a culture that stifles innovation, autonomy, and mastery. We don't optimize for the organization's purpose, the overarching goal. Instead, we reinforce resource efficiency, not flow efficiency.

Standardization—via policies and procedures—can strike anywhere in the organization. And, in my experience, the more policies and procedures you have, the more policies and procedures you create.

Back when I was a programmer, I had a boss who wanted to create coding standards. He'd heard all of us developers whining and complaining we couldn't read the code. He got tired of listening to the whining. He wanted us to be able to read each other's code.

He created coding standards—by himself—so he wouldn’t have to hear us complain anymore.

One Monday morning, he handed each of us a three-ring binder with at least 50 pages of coding standards. He’d tried to address all the particulars for this programming language—tabs vs spaces, how to name variables, and how long certain methods could be. We were supposed to follow these standards to the letter.

I was appalled. I looked around the office. Everyone else had the same look on their faces.

We all agreed something would have to be done. I was part of the six-person group who made an appointment with him. We walked in.

He smiled. “Is this a mutiny?”

In my ever career-limiting fashion, I said, “It might be.”

Everyone laughed. Including our boss.

We explained that since he wasn’t going to write code, he wasn’t the right person to set the standards. He could tell us the results he wanted. We would deliver those results.

My boss was happy we decided to stop whining and act. We created four pages of guidelines that would make one person’s code easier for others to read. While we didn’t all like everything, we used limited consensus to make sure we could live with everything. We all adhered to the guidelines. And yes, the code became much easier to read.

Coding standards are just one form of policy that assumes there is a single standard way to work. Coding *guidelines* allowed us to make it easy for other people to read the code.

Guidelines and constraints often produce better results than standardization.

## 13.1 Myth: I Can Standardize How Other People Work

Joseph, the CIO, smiled. “Okay, I’m really glad we can start this management meeting now. It’s time to talk about standardization. I want to create standards for our projects. I want to standardize on a single agile approach for all of our projects. I think you’ll all be pleased. Teams won’t be going off in every direction. We can standardize once and for all.”

Cathy, the QA director, wrinkled her forehead. “Uh, Joseph, are you telling us you want us to go ‘all in’ on that one agile approach right now?” she asked.

“Sure. Why not?”

“Well, we haven’t finished our pilot project, for one thing, and we don’t have enough money budgeted for training,” Dave, the Development Director, said. “And while I think an agile approach is a great way to go for many projects, our business counterparts have to think so, too. We need to bring them with us. Right now, they’re still thinking in six-month or year-long chunks. You can’t standardize on any agile approach without changing how they think.”

Joseph raised an eyebrow.

“Why do you care how we deliver, anyway, as long as we deliver effectively?” Cathy asked. “Our job is to solve problems. Your job is to make sure we are solving the right problems. If you decide which problems we solve by managing the project portfolio, we can decide how to solve the problems.”

Dave continued. “What if we decide that we need to prototype some architecture for a while to reduce technical risk? Are you going to have my head?”

Joseph looked at Dave for a minute, then said, “No, I’m not. But I thought you liked that agile approach.”

“I do,” Dave said. “But the developers and I don’t quite understand refactoring to patterns at the architecture scale that we have. We’re working at it.”

Cathy nodded. “That’s the same way my team doesn’t always understand how to create tests and refactor to test automation all in one iteration.”

“Transitioning to any agile approach—or any other approach—isn’t a slam dunk just because you declare it,” Dave continued. “It’s a change.”

“And why should we use just one approach?” Cathy asked. “Why shouldn’t we iterate on architectures or designs for a while if we want to? What’s wrong with that? And what about trying flow and WIP limits instead of formal iterations? Why can’t we do that? Why do we have to standardize on anything? Why can’t we experiment and see the results of our experiments? We need to learn from our pilot and experiments.”

Dave said, “I feel as if we are finally getting out of the yoke of waterfall. I don’t want to be back in the yoke of something I don’t understand. You hired me because I can think. I hired people because they can think. So did everyone else in this room. It’s time we let them think about *how* they do their work, not just *what* they do.”

“Forget the idea of standardization,” Cathy added. “Our projects are different from each other. Why should we use the same approach on each project?”

Joseph took a breath and looked as if he was about to say something.

Cathy held up her hand. “Let’s tell people the results we want and use a weekly or biweekly cadence to make sure we get the results in a reasonable amount of time. Why do we have to do more than that?”



Joseph leaned back in his chair. “Okay, as long as you reflect on your experiments and fix them when they go wrong, you have a deal.”

## 13.2 Standards Create a False Sense of Security

A “standard” approach to anything offers people—especially managers who don’t do that work—a false sense of security. They assume that a standard will force the work to proceed smoothly. How often does that happen? Not often enough!

You can standardize work on an assembly line and make the work safer and more efficient. But knowledge work? When you standardize knowledge work, you run the risk of making the work less innovative, less efficient, and not oriented to the real goal of your project.

Standardization for knowledge work might *look* efficient. However, standardization is often not effective. That’s because knowledge work has many unknowns and we can’t fully plan for those unknowns.

Since we can’t fully plan, we need to adapt our plans as we proceed based on more information. The best way to adapt the plans is to work with others in flow efficiency and create short feedback cycles to see our new reality. And, if teams have all the information they need, they can solve problems as a team.

As Don Reinertsen said in *The Principles of Product Development Flow: Second Generation Lean Product Development* [REI09](#):

“Decentralizing control requires decentralizing both the authority to make decisions and the information required to make these decisions correctly.”

If managers have all the authority and information, the teams can't make good decisions.

We have many principles for finishing work. When each team decides together, they can use the best of all the options. The team does need to know its overarching goal and all the necessary information to do great work.

When the team knows what they need to accomplish and they have control over their work, they have enough autonomy to improve their work.

### **13.3 Imposing a Standard Removes Autonomy**

Except for safety or regulatory requirements, I have yet to see a reason to impose a standard on someone else's work—especially if a manager does so. Even then, the people doing the work might see ways to be safe and live inside the regulations.

When managers impose standards, they implicitly say, "I don't trust you to do your jobs. Here. I will tell you how to work in detail." Do you want to do your job that way? I don't.

I actually like *my* constraints on my work. I like deadlines, as long as I can decide on the scope. For example, I often live with dates and word count when I write articles for other people. Not only can I live with constraints as constraints, I often find it a fun challenge to see what I can do inside those constraints.

There's a difference between telling someone how to do a job and providing constraints around the outcome.

When you tell people how to work, you might get malicious obedience. (Once, when I had to fill out a timesheet and limit the time to 40 hours, I stopped working overtime. My boss wasn't

happy, although I became much healthier because I paid attention to everything I had to do outside of work.)

When you explain the constraints, people can choose how to do the job.

But when managers told me how to do my job, I didn't live with that very well. I always thought of ways I could do it better. Always.

I often tried the work their way. Too often, it took me too long, or I couldn't get the outcomes we wanted. I finally got permission to do the work my way. I asked for guidelines and the necessary constraints. I got them.

Some of my managers were surprised by how well my approaches worked. That's because I was in the code or the tests or the project. I had the context. I knew the people. The managers were too far removed to be able to offer specific advice.

The people you lead and serve also know their work better than you do. That's their job.

I like to think about the work—the approach and the work itself. I bet the people you lead and serve do, also.

## **13.4 Policies and Procedures Prevent People from Thinking**

Managers create policies and standards to cover a multitude of past sins. Too often, when managers see “bad” or challenging behavior, they want a policy to prevent recurrence of that behavior. Instead, they could offer feedback about why that behavior doesn't work for the organization.

Worse, many standards try to cover all of the potential problems in a process. The standard wants to prevent people from thinking. That's how we got to big, honking binders of process.

The people who write the process binders read them. Other people rarely do. And, that's when we don't update the processes. We create brittle working systems.

## 13.5 Standards Create Brittle Systems

How much adaptability and resilience do you need in your organization? Adaptability is the ability to recognize a change and create alternatives to the current way we work. Resilience is the ability to capitalize on any of those alternatives.

Too often, standards create systems that discourage adaptability. The less often we practice adaptability, the less likely we are to be adaptable. And, the less often we try to change anything, the less resilience we have because we haven't practiced.

The more we try to mandate how people work, the fewer options people have to make choices. Fewer options means people experiment less often. Without learning from mistakes as well as successes, we can't build resilience in organizations.

Why do we hire people? To think and solve problems. Do we ever *not* want people to think? No. We want people to think. We want people to think hard. We want people to solve problems, whether it is with the process or the product.

We hired these people because we thought they were smart. They are. Let them show us how they apply their problem-solving skills to the project itself, not just the problem domain.

Sometimes, the people doing the work don't know how to solve the problem. That's when managers can go "meta" and address the environment that prevents people from solving the problem.