

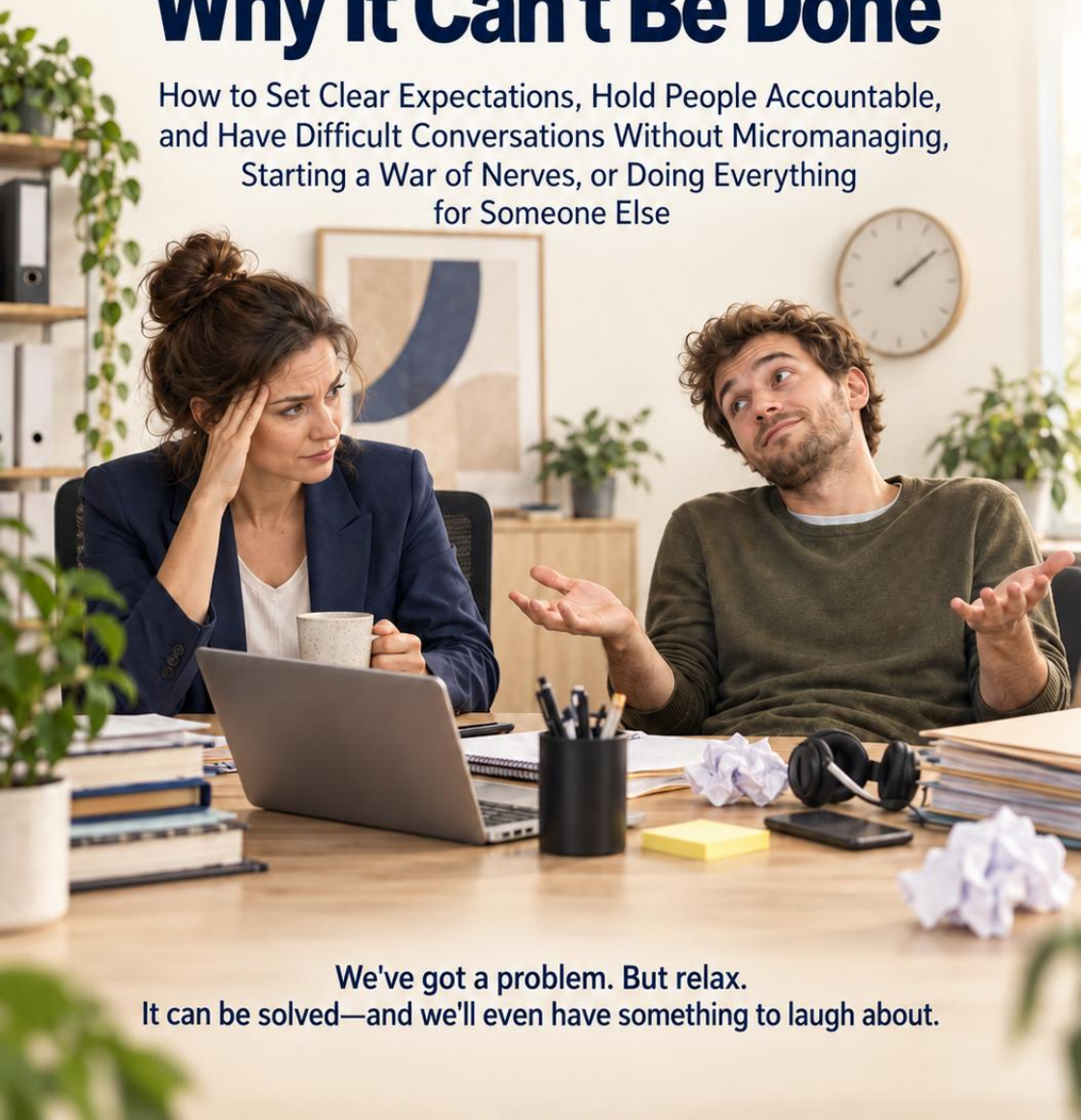
Max Paradox

PROBLEM? RELAX. YOU'VE GOT THIS!

A tongue-in-cheek guide series ;)

I Have an Employee Who Always Has a Reason Why It Can't Be Done

How to Set Clear Expectations, Hold People Accountable,
and Have Difficult Conversations Without Micromanaging,
Starting a War of Nerves, or Doing Everything
for Someone Else



**We've got a problem. But relax.
It can be solved—and we'll even have something to laugh about.**

I Have an Employee Who Always Has a Reason Why It Can't Be Done

How to Set Clear Expectations, Hold
People Accountable, and Have Difficult
Conversations Without Micromanaging,
Starting a War of Nerves, or Doing
Everything for Someone Else

Max Paradox

PROBLEM? RELAX. YOU'VE GOT THIS!

ALL-IN Rafał Płonkowski
2026
ISBN 9788384833803

CONTENTS

INTRO	5
Chapter 1 - A Reason Is Not the Same as Responsibility.....	11
Chapter 2 - Before You Hold Someone Accountable, Make Sure the Expectation Was Actually Clear	18
Chapter 3 - When "It Can't Be Done" Becomes a Habit	26
Chapter 4 - The Helpful Manager Who Keeps Rescuing Everyone.....	34
Chapter 5 - Stop Putting Every Excuse on Trial	43
Chapter 6 - Don't Check More Often. Check at Better Moments.....	51
Chapter 7 - When Everything Is Urgent, Accountability Has Somewhere to Hide	59
Chapter 8 - Stop Protecting Employees from the Consequences of Their Own Decisions	66
Chapter 9 - Turn "I'll Try" into a Real Commitment	74
Chapter 10 - Have the Difficult Conversation Without Turning It into an Indictment.....	82
Chapter 11 - Build an Escalation Ladder Before the Problem Lands on Your Desk	90
Chapter 12 - Measure Results, Not the Performance of Looking Extremely Busy	98
Chapter 13 - Is This an Accountability Problem or a Capability Problem?.....	106
Chapter 14 - Decide What Your Employee Is Allowed to Get Wrong Without Asking You First.....	114

Chapter 15 - When You've Had the Same Conversation Five Times and Nothing Changes	122
Chapter 16 - You Cannot Coach Every Person into the Right Role.....	131
Chapter 17 - Check Whether the Problem Is Really the Employee or the System Around Them	140
Chapter 18 - Build Accountability That Still Works When You Stop Watching.....	148
Chapter 19 - Don't Wait Until You Hear "It Couldn't Be Done" Again.....	157
Chapter 20 - The Best Ending Is When You Are No Longer Needed for Every Decision	165

INTRO

It is 9:14 on a Tuesday morning, and you ask a perfectly ordinary question: "Where are we with the client proposal?" You are not angry. You are not standing under a flickering interrogation light. Nobody has been offered immunity in exchange for testimony. You simply want to know whether the thing that was supposed to be finished yesterday is, in fact, finished. What follows is a remarkably complete explanation of why it is not. The pricing team answered late, someone changed a requirement, the shared folder contained the wrong version, another department needed something urgent, and there was apparently a brief period on Monday afternoon when every available route to progress was closed by forces normally associated with maritime disasters. Five minutes later, you understand the history of the problem in exquisite detail. What you still do not have is the proposal.

The maddening part is that some, perhaps all, of those reasons may be true. Colleagues do reply late. Systems fail. Customers change their minds after previously demonstrating tremendous confidence in the mind they had yesterday. Priorities collide, managers contradict each other, and organizations occasionally design approval processes that appear to have been created as a practical demonstration of endurance. The problem is not that an employee has encountered an obstacle. Good employees encounter obstacles all the time and tell you about them. The problem begins when the obstacle quietly becomes the end of ownership: "I couldn't complete it because X happened" starts to mean "once X happened, my responsibility

stopped.” That is the moment when an explanation stops being useful context and begins functioning as an exit door.

Managers often make this problem worse for entirely understandable reasons. You want the work finished, so when an employee arrives carrying an unresolved problem, you help. You know whom to call, where the information lives, which exception can be approved, and which sentence will get the difficult stakeholder to answer. You solve the issue in ten minutes, feel briefly efficient, and go back to your own work. Then another issue arrives. And another. Eventually, you have built an extremely dependable workflow in which your employee encounters obstacles and you remove them. The employee becomes highly skilled at bringing you problems at the correct stage of incompleteness, while you become a combination of manager, emergency service, search engine, diplomatic envoy, and mysterious person who apparently has access to a secret button marked MAKE THIS HAPPEN.

When helping does not work, many managers move to the opposite strategy: more control. You ask for updates earlier, then more often, then before lunch, then after lunch because something may have changed during the dangerous forty-five-minute period in between. You request to be copied on messages. You start checking drafts that do not yet need checking. Soon you are attending meetings whose only purpose appears to be confirming that the meeting scheduled to discuss progress will still take place. The logic is obvious: if unpleasant surprises keep appearing, more visibility should prevent them. Unfortunately, excessive control can create the exact dependency you are trying to eliminate. If the manager will check everything anyway, the employee has less reason to build their own reliable system.

If every decision will be reviewed, asking becomes safer than deciding. You gain information and lose autonomy at the same time.

The third common response is the Big Conversation. You save up frustration until you finally sit the employee down and explain everything that has been going wrong. Not just yesterday's missed deadline, of course. Once the gates are open, March would like a word. So would April. There was also that thing before the holiday, the meeting where nobody had the numbers, the email that somehow became three days of silence, and the task that was "basically done" in the same way that an empty building site is basically a house if you are optimistic enough. The employee starts explaining the circumstances behind each example, you explain why the circumstances do not fully excuse it, and within twenty minutes you are both conducting an archaeological excavation of the employment relationship. A difficult conversation may be necessary. But a successful difficult conversation is not the one in which the manager finally says everything. It is the one after which something observable changes.

This book is about creating that change without turning yourself into either a full-time rescuer or a workplace surveillance program. It is not about teaching employees never to say "I can't." Sometimes something genuinely cannot be done under the current conditions, and pretending otherwise is not accountability but corporate theatre with a deadline. Nor is it about treating every problem as evidence of laziness, bad attitude, or moral weakness. An employee may lack authority, skill, information, capacity, or a functioning process. A manager may have given vague instructions, changed priorities repeatedly, or unknowingly

trained the team to bring every difficult decision upward. The useful question is therefore not "Who can I blame for this?" but "Where did ownership stop, and what needs to change so that it does not stop there next time?"

You will learn to make expectations concrete enough that both people are actually agreeing to the same job, not two private versions of it. You will learn to separate a genuine external obstacle from the employee's response to that obstacle, because those are different things and should be managed differently. You will see how to define decision rights, escalation points, commitments, consequences, and checkpoints without constructing a bureaucracy so impressive that completing the task becomes secondary to reporting on it. You will also learn when an employee needs coaching, when they need clearer boundaries, when they need more authority, and when repeated explanations are no longer a communication problem. Most importantly, the goal is not to create a person who never needs you. It is to create someone who needs you for the things a manager should actually be needed for.

That requires a change in one of the most seductive habits of competent managers: doing things because you can do them faster. You probably can. You may write the message better, handle the awkward conversation more smoothly, spot the flaw sooner, and know exactly which shortcut is safe. That is why rescuing feels so rational. But every time you automatically take over a difficult piece of someone else's role, you solve today's problem at the possible expense of tomorrow's capability. The alternative is not to stand back coldly while everything burns in the name of personal development. It is to help in a way that leaves ownership where it belongs: provide the missing information,

make the decision only you can make, remove the barrier only you can remove, and then hand the work back.

There is also an uncomfortable promise hidden inside better accountability: once expectations become clearer, excuses become less interesting. You no longer need to spend fifteen minutes deciding whether the reason was "valid enough." You can acknowledge the circumstance and still ask what happened next. You can accept that another department caused the delay while examining why nobody raised the risk earlier. You can understand that the week became overloaded while still asking when the employee recognized the conflict and what they did about it. This makes conversations less personal and, strangely, often less hostile. Instead of arguing about whether somebody is responsible as a human being, you can discuss a specific responsibility in a specific situation. That is a much smaller battlefield.

The standard we are aiming for is not perfection. Responsible employees miss things, make poor judgments, misunderstand information, and occasionally discover that their brilliant plan was only brilliant before contact with reality. Responsible managers do the same. The difference is what happens after the problem appears. Does the employee disappear behind the reason, or remain engaged with the outcome? Do they bring only the obstacle, or also the work they have already done around it? Do they wait for rescue, or know when to act, when to ask, and when to escalate? And can you, as the manager, resist the very human urge to either take over everything or make the employee regret ever bringing you bad news?

If you can build that kind of working relationship, the most important change will be surprisingly quiet. Fewer tasks will

boomerang back to your desk. Difficult conversations will become shorter because expectations will already be visible. Problems will still happen, because apparently no management method has yet persuaded reality to cooperate permanently. But the sentence "there was a problem" will no longer automatically be followed by "so nothing could be done." It will be followed by something far more useful: what happened, what was tried, what decision is needed, and what happens next. That is the difference between managing reasons and managing responsibility.

Chapter 1 - A Reason Is Not the Same as Responsibility

The first time you hear a good reason for a missed result, you usually listen. The fifth time, you begin listening differently. You are no longer hearing a description of reality; you are mentally searching for the small trapdoor through which responsibility is about to leave the room. "The vendor did not respond." "Finance changed the numbers." "The customer moved the meeting." "The software was down." Each statement may be perfectly accurate, which is precisely what makes the situation difficult. If the employee were simply inventing nonsense, management would be easier. Unfortunately, most workplace excuses are built from real events, and real events are stubborn things to argue with. The useful distinction is therefore not between truth and fiction. It is between explaining what happened and taking responsibility for what happened next.

Imagine that an employee was supposed to confirm a shipment by Wednesday. On Thursday you discover that nothing has been confirmed because the supplier never replied to Monday's email. The employee has not lied. The supplier genuinely did not reply. But between Monday's email and Thursday's discovery sat several possible decisions: send a follow-up, call, try another contact, notify you that the deadline was in danger, prepare an alternative supplier, or at least decide when waiting would stop being an acceptable strategy. The missing reply is the reason the original plan became difficult. It is not, by itself, a complete explanation of why no further action occurred. Responsibility

lives in that gap. It begins where circumstances stop cooperating and somebody still has to decide what to do.

Managers often blur this distinction because they ask the wrong question first. "Why wasn't this done?" sounds sensible, but it invites a history lesson. Once the employee starts explaining, the conversation naturally follows the chain of causes backward: who failed to send what, when the system went down, which meeting overran, whose approval was missing. Ten minutes later you may understand the ecosystem beautifully while knowing very little about the employee's decisions. A better question is, "What did you do when you realized the original plan was no longer working?" That question does not deny the obstacle. It simply moves attention from the weather to the steering wheel. You are no longer asking whether the storm existed. You are asking what happened after somebody saw the clouds.

This matters because accountability does not mean controlling everything. An employee cannot force another department to answer. They cannot repair a corporate platform by concentration, persuade a customer not to change their mind through spiritual discipline, or manufacture budget authority they do not possess. A manager who says "you own the outcome" can accidentally demand supernatural powers under a fashionable business phrase. Real accountability is narrower and more useful. The employee owns the actions that reasonably sit within their role: noticing the issue, communicating it at the right time, taking available steps, making decisions within authority, and escalating when the situation genuinely requires a different level. They may not control the obstacle, but they can often control their response to it.

This is also where two managerial mistakes appear. The first is accepting every external obstacle as a full release from responsibility. "Oh, Legal did not answer? Fair enough." Conversation over. Do that often enough and you train people to treat the first dependency as a finish line. The second mistake is dismissing every obstacle as an excuse. "I don't want to hear reasons, I want results." That may sound wonderfully decisive for approximately seven seconds. Then employees learn that bringing you bad news is dangerous, so they delay it until the problem has matured into something large enough to have its own meeting invitation. Neither extreme produces ownership. One creates passivity; the other creates concealment.

A more useful conversation separates three questions that managers frequently mash together. First: what happened? Second: what part of that was within the employee's influence? Third: what action did the employee take inside that area of influence? Suppose a client unexpectedly changes a requirement two days before delivery. The change itself is not the employee's fault. The need to reassess scope is now part of the employee's job. If they immediately review the impact, tell you the deadline is at risk, and bring two options, they are acting responsibly even if the final delivery moves. If they say nothing, continue hoping, miss the deadline, and then explain that "the client changed everything," the same external event has produced a very different level of accountability.

You do not need a courtroom tone to uncover the difference. In fact, a prosecutorial style usually makes the conversation worse because it turns the employee into a defense attorney. Ask when they first knew there was a problem. Ask what they tried. Ask what options they

considered. Ask when they decided the original target was no longer realistic. Ask whom they informed. These questions reconstruct decisions without requiring you to accuse anyone of “making excuses,” a phrase almost guaranteed to trigger a secondary debate about the moral quality of the explanation. Once you label something an excuse, the employee feels compelled to prove that the obstacle was real. Once you focus on behavior, the obstacle can be completely real and the response can still be judged separately.

Consider two employees facing the same problem. Both are waiting for numbers from a colleague who is unexpectedly absent. Employee A sends one message and waits. By the next afternoon, the numbers have still not arrived, so A tells you the report cannot be completed because the colleague is out. Employee B sends the same message, notices the absence, checks whether the information exists elsewhere, contacts the colleague’s backup, completes the sections that do not depend on the missing data, and warns you that the final section may slip. The obstacle is identical. The outcome may even be identical if neither employee gets the numbers. What differs is ownership. One employee treated the dependency as the end of action. The other treated it as a change in the problem to solve.

This distinction is especially useful because it protects employees from unfair blame while making expectations stronger. If you only measure whether the final result happened, people can be punished for events genuinely outside their control. If you only listen to circumstances, people can avoid responsibility by describing the environment in sufficient detail. Looking at the response

gives you a fairer middle ground. You can say, "I agree that you could not control the supplier's delay. What I expected from you was an earlier warning and an alternative plan." That is far more precise than "you need to take more ownership," which may sound impressive but leaves the employee wondering what to physically do the next time somebody fails to answer an email.

There is another uncomfortable part: managers sometimes train employees to stop owning problems by solving them too quickly. An employee says, "I cannot finish because I need information from Operations." Before they have completed the sentence, you are already typing to the head of Operations. Two minutes later, you have found the information, copied the employee, and felt appropriately useful. The immediate business problem is solved, but a learning event has also occurred. The employee has learned that bringing you an unresolved dependency is an efficient strategy. If this happens repeatedly, you may later complain that they "always come to me with problems," while forgetting that your response has made this behavior extraordinarily rewarding. Fast rescue can be excellent for the deadline and terrible for the pattern.

A simple change is to delay your own solution long enough to hear theirs. When an employee brings a blocker, ask, "What have you tried so far, and what do you think the next move should be?" If they have a sensible plan and need your authority for one part, great. Make the decision only you can make and leave ownership of the rest with them. If they have no idea, do not automatically turn the conversation into a guessing game designed to prove their independence. Sometimes people genuinely lack experience. You can help them think through two or three

possibilities, but finish by identifying which action remains theirs. Support should increase the employee's ability to move the work forward, not quietly transfer the work to the manager.

The minimum version of this method requires no framework, form, dashboard, training session, or laminated card bearing the word ACCOUNTABILITY. The next time you hear, "I couldn't because...", do not immediately challenge the reason and do not immediately solve the problem. Ask one question: "What did you do after that happened?" Then listen. If the answer contains several sensible actions, you may be dealing with a real constraint rather than avoidance. If the answer is effectively "I waited," you have found something concrete to discuss. From there, ask what the next reasonable action could have been and agree on how similar situations should be handled in the future.

The Plan B is for the employee who answers every question by throwing it gently back into your lap. "I don't know. What do you want me to do?" If the decision genuinely belongs to you, make it. But if it sits within their role, resist becoming the automatic answer machine. Give them a boundary instead: "Bring me two reasonable options and tell me which one you recommend." If they do not have enough knowledge to generate those options, teach the decision once by explaining the criteria you use. The next similar situation should require less help. The point is not to make asking for help embarrassing. The point is to make sure help does not become a permanent substitute for thinking.

Over time, this changes the language of the relationship. Instead of hearing only "the vendor did not reply," you begin hearing "the vendor did not reply, so I contacted the backup

and moved the unaffected part forward." Instead of "the system was down," you hear "the system was down for two hours, so I switched to the manual process and flagged the remaining delay." Instead of "I couldn't," you hear a description of what was still possible. That is the shift you want. Not a world without obstacles, but a workplace in which obstacles do not automatically cancel ownership.

The practical test is simple. The next time something slips, spend less time deciding whether the reason is good enough and more time examining the response. Ask when the obstacle became visible, what action followed, what remained within the employee's control, and whether the right people knew early enough. If the employee acted sensibly, recognize that and address the external problem. If they stopped at the obstacle, make the next expected behavior explicit. Responsibility is not the absence of reasons. It is the refusal to let a reason become the point where useful action quietly ends.

Chapter 2 - Before You Hold Someone Accountable, Make Sure the Expectation Was Actually Clear

A manager says, "Can you put together an update for Friday?" and walks away believing an agreement has been made. In the manager's head, "update" means a concise summary of progress, major risks, revised milestones, three key numbers, and a recommendation that can be presented to senior leadership at 10:00 a.m. Friday. In the employee's head, "update" means a paragraph describing what happened this week, probably sent sometime Friday afternoon. Both people have heard the same sentence. Both think it was clear. On Friday, the employee proudly delivers six lines of text, and the manager stares at them with the expression of somebody who ordered a dining table and received a surprisingly attractive coaster. This is the moment when many managers conclude that the employee is careless. Sometimes they are. Sometimes the assignment existed in two completely different versions from the beginning.

Accountability becomes unfair when expectations are visible only inside the manager's head. The more experienced you are, the easier it is to underestimate how much context you are carrying. You know why the work matters, what the executive team usually asks, which mistakes caused trouble last quarter, what "good enough" looks like, and which details can be ignored. The employee hears the words you actually say, not the background music playing in your brain. You may genuinely believe that "prepare the figures for the review" contains all necessary