Solve your problem with a wink

My Boss Told Me to Automate Something Nobody Understands

How to Organize a Process Before Implementing AI and Measure the Real Gains



Max Paradox

My Boss Told Me to Automate Something Nobody Understands

How to Organize a Process Before Implementing AI and Measure the Real Gains

Max Paradox

PROBLEM? RELAX. YOU'VE GOT THIS!

ISBN: 9788384905340

ALL-IN Rafał Płonkowski



CONTENTS

INTRO 5

Chapter 1 - Before You Switch On AI, Figure Out What Is Actually Supposed to Work 11

Chapter 2 - The Process According to the Presentation Versus the Process According to the People Doing the Work 19

Chapter 3 - Not Every Problem Needs Artificial Intelligence, Even If Your Boss Has Already Bought a License 29

Chapter 4 - Automating Chaos: How to Spend Money Making Mistakes Faster 38

Chapter 5 - Data Is Everywhere, but the Truth Is Nowhere to Be Found 48

Chapter 6 - People Aren't Afraid of Automation. They're Afraid of What the Company Will Do to Them Afterwards 57

Chapter 7 - The Project That Was Supposed to Do One Thing and Is Now Expected to Save the Entire Company 66

Chapter 8 - The Never-Ending Pilot: Artificial Intelligence That Works Brilliantly Until Someone Actually Uses It 76

Chapter 9 - From Chaos to Instructions Even a Computer Can Understand 86

Chapter 10 - How Much Does Automation Really Earn When We Stop Believing the Presentation? 95

Chapter 11 - Test AI as Though It's Going to Make a Mistake. Because Sooner or Later, It Will.....105

Chapter 12 - Human in the Loop: Who Has the Authority to Tell AI to Stop?..... 114

Chapter 13 - The Big Launch Day, or Why Monday at 8:00 a.m. Is a Terrible Time for a Revolution.....124

Chapter 14 - The Dashboard Says Success. But How Do We Know AI Deserves the Credit? 133

Chapter 15 - When AI Fails and Everyone Pretends It's Just a Minor Inconvenience 143

Chapter 16 - When Automation Doesn't Deliver Savings, Even Though Everyone Tried Really Hard 152

Chapter 17 - We Have No Budget, No People, and No Time: Automation in a Real Company..... 161

Chapter 18 - Automation Isn't Furniture. You Can't Just Install It and Forget About It.....170

Chapter 19 - When Automation Becomes a Sacred Cow: How Not to Lose the Courage to Simplify 180

Chapter 20 - The Next Time Your Boss Says "We're Implementing AI," You'll Know What to Do 189

INTRO

It's Monday, 9:07 a.m., and your boss walks into the meeting looking like someone who has just discovered the future. He announces that the company needs to accelerate its digital transformation, harness artificial intelligence, and eliminate inefficiencies. Everyone nods. Someone writes "AI" in a notebook, someone else opens a presentation, and you wonder whether it would be appropriate to ask what exactly we're planning to automate. It wouldn't. You're a professional, after all. Professionals don't ruin the atmosphere by asking questions nobody knows how to answer.

Twenty minutes later, a decision has been made. The reporting process needs to be automated. The deadline? Preferably yesterday. The budget? To be determined. The process owner? Probably everyone, which in practice means no one. Documentation? It's somewhere on the shared drive, although the last confirmed sighting dates back to when the office printer still worked without IT intervention. The success must be spectacular, measurable, and something the management board can appreciate in three slides. A fourth slide would be unnecessary bureaucracy.

You've just been handed an assignment that sounds modern, looks prestigious, and has all the ingredients needed to become an expensive mess.

The funniest part is that the problem doesn't actually begin with artificial intelligence. It starts much earlier, when five people perform the same task in five different ways, three spreadsheets contain five versions of the truth, and the only person who understands how they all fit together is

Marek from controlling. Marek has been on holiday for two weeks and isn't answering his phone. The company suddenly discovers that its most important information system is Marek.

Under these circumstances, implementing AI is rather like hiring an incredibly fast assistant whose responsibilities nobody can explain. The assistant can analyze data, produce reports, identify patterns, and prepare answers. What it cannot reliably guess is why one department considers an order complete when the invoice is sent, while another considers it complete only after payment arrives. If you ask it to accelerate chaos, you may end up with chaos at an impressive speed. Now with automatic notifications.

That doesn't mean artificial intelligence is overrated. Quite the opposite. It can be extraordinarily useful. The problem arises when an organization treats it as a miracle cure for everything it previously couldn't be bothered to put in order. Instead of identifying where errors originate, it buys a tool to process them faster. Instead of clarifying responsibilities, it creates a digital assistant expected to cooperate with people who are convinced responsibility belongs to another department. Instead of simplifying a procedure, it automates every twist, exception, and historical oddity that has accumulated along the way.

And this brings us to a question that is often missing from project meetings: does this process even deserve to be automated in its current form?

Perhaps half its steps are unnecessary. Perhaps a report that takes six hours a week to prepare hasn't supported a single decision in months. Perhaps three people check the same figures because one mistake occurred years ago, and the organization responded by effectively assigning an

entire department to make sure it never happened again. Automation can help, but first we need to establish what genuinely creates value and what merely creates the appearance of work.

In this book, we won't treat technology as a religion or vendor presentations as divine revelations. Nor will we pretend that every employee should become a programmer, a data analyst, and a language-model expert, preferably before Thursday. Instead, we'll tackle something considerably more down-to-earth and, at the same time, more difficult: understanding the work we want to change.

You'll discover how to identify where a process really begins and ends, even when the people involved describe it as though they work for entirely different companies. You'll learn to distinguish necessary activities from those performed purely out of habit, identify where mistakes arise, and establish who makes which decisions. We'll also examine situations in which information circulates between messages, spreadsheets, and systems while a human being serves as a biological USB cable.

One particularly important distinction involves three things that corporate meetings love to bundle together under the convenient label "improvement." Organizing a process means establishing clear rules and eliminating unnecessary activities. Conventional automation allows repetitive tasks to be performed according to defined rules. Artificial intelligence can provide additional support when interpreting less structured information, classifying material, or preparing suggestions is required. These approaches can complement one another, but they are not interchangeable. If a simple formula or system rule will do the job, bringing in an AI

model may be the equivalent of ordering a helicopter to move a binder from one desk to another.

Money will be the other central character in this book. Not the imaginary money displayed on a slide entitled "Potential Strategic Benefits," but actual costs and savings. Suppose an employee spends ten hours a week on a task, and a new solution reduces that time to two. Sounds excellent. Has the company saved eight hours? In terms of the work required for that task, possibly. Has it saved the equivalent of eight hours of wages? Not necessarily. The employee still receives a salary, the new tool costs money, someone has to check the results, and maintaining the solution also requires work.

The real benefit depends on what the organization does with the time it recovers. It might handle more cases, reduce overtime, avoid additional hiring, or improve service quality. Alternatively, it might give everyone more time to wonder why the new system generates a table with an empty column every Friday. Simply multiplying hours by an employee's hourly rate is not enough to demonstrate financial success.

That's why we'll learn to establish a baseline before implementation begins. How long does the work take? How many cases come back for corrections? How often do exceptions occur? What does handling errors cost? Which activities will genuinely disappear, and which will merely be transferred to someone or something else? Without this information, it's easy to present an improvement that cannot actually be verified. It's rather like announcing that you've lost five kilograms before stepping onto a scale for the first time. The result looks fantastic, especially in Excel.

We'll also keep the limits of automation firmly in view. Not all data can be shared with just any tool, not every AI-generated answer is correct, and not every decision should be left to a system without human oversight. In some processes, a mistake costs an additional five minutes of work. In others, it can cause financial losses, breaches of confidentiality, or serious consequences for customers. A good implementation recognizes that difference rather than hiding it behind a colorful savings chart.

You don't need to be a director of transformation or have influence over every decision in the company. Sometimes all you can do is organize one part of your team's work, gather some data, and present a sensible proposal. That's enough to start making changes without promising a revolution. And if your boss expects a spectacular implementation within two weeks, you'll find arguments that help you discuss scope, risks, and expected benefits without resorting to the phrase "This will never work," which rarely earns a standing ovation at a management board meeting.

What matters most is that, by the time you finish reading, you'll be able to take a vague instruction and turn it into a series of sensible decisions. You don't need to know every answer immediately. You should know which questions to ask, what to investigate, and when to say that there isn't yet enough evidence to choose a technology.

Because artificial intelligence can accelerate an astonishing number of things. Understanding the problem, however, is not a step worth skipping.

And the next time your boss announces that AI must be implemented immediately, perhaps instead of nervously opening a vendor directory, you'll calmly ask:

"Great. But could we first establish what it is we're actually doing?"

That might turn out to be the most productive sentence of the entire project.

Chapter 1 - Before You Switch On AI, Figure Out What Is Actually Supposed to Work

A meeting appears on the company calendar under the title "Process Optimization Using AI." Eleven people have been invited, including three managers, two IT specialists, and someone from finance who is probably there because somebody mentioned the word "savings." The agenda consists of a single sentence: "Discuss automation opportunities." The meeting lasts an hour. For the first twenty minutes, everyone talks about how dramatically artificial intelligence will change the future. During the next twenty minutes, they name various tools. In the final few minutes, they agree that a process analysis needs to be prepared. Unfortunately, nobody can explain where the process begins, where it ends, or who is responsible for its outcome.

Congratulations. You've just attended the first project meeting. Three action items, four additional meetings, and a presentation have been created. The problem itself remains untouched, but at least it now has a professional-sounding name.

This scenario is fictional, although the mechanism behind it is hardly unusual. Organizations often begin technology projects by discussing solutions because solutions are more attractive than problems. A new tool can be demonstrated on a screen, described with an impressive slogan, and included in a strategy document. Analyzing how work actually gets done, on the other hand, requires talking to

people who perform tasks that management rarely sees. It means asking uncomfortable questions, observing exceptions, and acknowledging that some corporate procedures exist mainly because nobody has found the time to challenge them.

Let's begin with a fundamental distinction. A process is not the name of a department, a job title, or a computer program. It is a sequence of connected activities that transforms specific information, materials, or decisions into a concrete result for a recipient. It can be described, observed, and evaluated. When someone says the process is "customer service," they're really naming an area of business activity. When they say, "From receiving a customer complaint to communicating the decision and closing the case," we begin to understand what we're looking for.

The difference may seem insignificant until you attempt to automate all of "customer service." Suddenly, the same project includes receiving inquiries, answering questions, handling complaints, updating records, preparing documents, and managing a dozen exceptions. The initiative becomes so broad that almost anything could be included, even the coffee machine in reception, provided it reports a malfunction through the customer service form.

The first task, therefore, is to narrow the scope.

Imagine a distribution company where a manager wants to automate the preparation of a weekly sales report. The team downloads data from a system, fills in a spreadsheet, compares results against the plan, investigates discrepancies, and sends the report to management. Employees complain that reporting takes too much time. The manager believes AI should prepare the report

independently. A technology vendor promises that its solution can analyze data and generate commentary. Everything looks promising.

Only when someone speaks to the employees preparing the report does the real situation become clear. Downloading the data takes just a few minutes. Most of the time is spent figuring out why the figures from two systems differ, manually assigning sales to the correct categories, and correcting data whose definitions are inconsistent. In other words, the real problem isn't writing the report. It's getting the information into a condition where people can trust it.

If the company implements an AI commentary generator, it will receive explanations of potentially incorrect data more quickly. The result will be an elegant document containing the wrong figures. Technology won't have solved the underlying problem. It will simply have given it a more attractive layout.

Where Does the Real Process Begin?

A useful way to investigate a process is to follow one specific case from beginning to end. Don't ask only how the procedure is supposed to work. Ask an employee to show you the most recent real example. When did they receive the information? What did they do first? How did they know they could proceed? Who received their output? At what point did they consider their task complete?

This approach protects you from something we might call the official version of reality. According to the documentation, everything happens in a straight line. The system receives data, an employee verifies it, a manager approves the result, and the recipient receives the information. In practice, the employee phones a colleague

because the system displays an outdated status. The colleague checks another system, sends a message to the warehouse, and waits for a reply. The manager approves the result only after receiving an additional report that doesn't appear in any instruction manual.

When you ask why everyone performs these extra activities, you'll probably hear, "Because there's no other way." That's an important answer, but it isn't the end of the investigation. It might indicate a genuine technical limitation, missing access permissions, unreliable data, or a rule whose original purpose has long since been forgotten. Each of these causes requires a different solution.

During the initial observation, it is enough to record five things:

1. What triggers the process, and what result should it produce?
2. Who performs each activity, and who is responsible for the final outcome?
3. What information and systems are required?
4. Where do waiting, corrections, exceptions, and unclear decisions occur?
5. How does the recipient know that the case has been completed correctly?

You don't need to create a professional diagram featuring twenty different symbols immediately. A sheet of paper, a simple flowchart, or a table is often enough to reveal the first connections. What matters most is recording what people actually do, rather than an idealized version of the procedure. If an employee copies data between systems every morning because no integration exists, include that

activity. Don't erase it simply because it looks embarrassing in a presentation.

Let's return to the sales report. Suppose an analyst downloads the data on Monday morning. She then waits for updates from two teams, manually checks product classifications, removes duplicates, and explains discrepancies. Only then does she prepare the report. Her manager reads the document and requests an additional breakdown of the results. The report goes back for corrections before finally reaching its intended recipients.

In the official description, the process has three steps: download the data, prepare the report, and send it. In reality, it contains a dozen or more activities, several decision points, and at least two places where work can stop for an indefinite period.

This doesn't mean every one of those activities needs to be automated. It simply means that, at last, we know what we're dealing with.

When Everyone Is Right but the Process Still Doesn't Work

One of the most deceptive problems occurs when everyone performs their individual part of the work correctly, yet the overall process delivers a poor result. The sales department provides information according to its schedule. Finance requires more detailed verification. Operations relies on updates that become available only the following day. Everyone has a reasonable explanation. Nobody feels responsible for the time lost between these activities.

That's precisely why asking people whether they're performing their tasks correctly isn't enough. You need to establish whether the output of one activity is sufficient for

the next person to begin working without further explanations. A process can consist of individually correct activities and still be an organizational disaster. The departments then resemble musicians flawlessly performing different pieces of music at the same time. Each deserves applause, but the audience leaves.

In practice, it helps to distinguish responsibility for an individual activity from responsibility for the outcome of the entire process. The person preparing the data may be responsible for its completeness, the manager for approving it, and the process owner for ensuring that the whole operation meets agreed time and quality standards. In a small team, one person may perform several roles. What matters is that responsibilities are explicit rather than determined only after something goes wrong.

You won't always have the authority to change how responsibilities are distributed. You can, however, expose a gap. Instead of saying, "Nobody is responsible for this," present a specific observation: "After data is transferred between departments, there is no designated person monitoring discrepancies and deciding how they should be resolved." This statement allows people to discuss the necessary action rather than search for someone to blame. It's particularly useful when project participants occupy different positions and each has a well-prepared collection of defensive arguments.

It's also worth distinguishing a process problem from an isolated execution error. If one employee mixes up columns in a spreadsheet, the cause might simply be a momentary lapse in attention. If several employees repeatedly confuse the same columns, you should examine the form's design, terminology, and information handover procedures.

Automation can be an effective safeguard against recurring errors, but only once you understand where those errors originate. Otherwise, you may create a system that consistently selects the wrong column. It will do so without getting tired, taking holidays, or experiencing the slightest pang of guilt.

Don't Confuse a Symptom with a Cause

"Reporting takes too long" is an observation. "Employees are too slow" is an interpretation. "Two systems use different definitions of the sales date, so every update requires manual reconciliation" is a hypothesis about the cause that can be tested.

This distinction changes the quality of the discussion. Instead of wondering how to make an employee work faster, you can investigate whether definitions are consistent, whether data is available, and how many manual corrections are required. Sometimes the most effective improvement won't be an intelligent agent but agreeing on a single way of calculating a result. It won't look particularly impressive at a technology conference, but it may save many hours of arguments.

Asking "why?" several times can be useful, provided you don't turn the conversation into an interrogation. Why is the report late? Because we're waiting for data. Why are we waiting? Because it arrives after manual verification. Why is verification necessary? Because the figures in different systems sometimes disagree. Why do they disagree? Because the systems update at different times and classify certain transactions differently.

At this point, the problem becomes specific. You can investigate the scale of the discrepancies, agree on rules,

and only then decide which activities are worth automating. Don't assume, however, that every additional "why?" leads to one ultimate root cause. Processes often have several difficulties operating in parallel. Sometimes the problem involves poor data quality, missing decisions, and an unrealistic deadline all at once.

At the end of your initial investigation, try writing a single sentence describing the problem without using the words "AI," "automation," or "transformation." For example: "Every week, we lose time manually reconciling data from two systems because their definitions and update schedules differ." This sentence doesn't determine how the problem should be fixed, but it identifies an area that can be investigated.

If you can't formulate such a sentence, you need more observation, not a more expensive tool.

In a well-managed project, the first success doesn't necessarily have to be a functioning automated solution. It may be the moment when the participants agree, for the first time, on the problem they're actually trying to solve. That sounds modest, but it's enormously important. Only then does technology stop being an answer invented before anyone has asked the question.

And if someone at the next meeting asks when you're finally going to start implementing artificial intelligence, you can calmly reply, "We're establishing what it should do so we don't end up automating our own mistakes."

That's a considerably better beginning than purchasing licenses and collectively hoping for the best.

Chapter 2 - The Process According to the Presentation Versus the Process According to the People Doing the Work

There's a diagram on the screen. Five rectangles, four arrows, and one diamond representing a decision. Everything is perfectly aligned, the colors are consistent, and the design is so elegant that you might conclude the company operates without the slightest friction. The first rectangle says "Receive request," the second "Verify data," the third "Approve," the fourth "Execute," and the fifth "Close case."

The manager presents the diagram with pride. Then he asks an employee who handles requests whether the process really looks like that.

"Yes, more or less," she replies.

In corporate language, "more or less" can contain an astonishing amount of information. In this case, it includes an additional spreadsheet, seven types of exceptions, two unofficial messaging groups, and manual data checks because the system occasionally displays a status it shouldn't. The diagram is still beautiful. Unfortunately, it describes an organization that exists mainly during presentations.

This doesn't necessarily mean the diagram's creator was trying to deceive anyone. Procedures often describe the intended way of working rather than every situation encountered in practice. They're created during system implementations, audits, or reorganizations. Later, customers, requirements, products, and employees change.

New exceptions, workarounds, and informal arrangements appear. The document remains untouched because updating it rarely makes the list of urgent tasks. As a result, the organization gradually develops two processes: the official one, which can be presented, and the actual one, which allows the work to get done.

For an automation project, this distinction is fundamental. A tool designed exclusively around the official procedure may perform brilliantly during a demonstration and fail the moment it encounters a genuine request. Not because the programmers did a poor job. They simply automated a world where every form is complete, every piece of data is up to date, and every decision is made exactly when scheduled.

Such a world can sometimes be found in a test environment. It generally doesn't have any customers.

What You Won't See on the Diagram

Imagine a department responsible for processing business orders. The official process involves receiving an order, checking availability, confirming the terms, and forwarding the documents for fulfillment. Simple enough. During a conversation, however, an employee explains that some customers send orders by email, others use a form, and still others communicate changes over the phone. One customer regularly enters outdated product codes. Another orders products in combinations the system doesn't recognize.

Each case requires interpretation. The employee understands the history of the business relationship, can recognize mistakes, and knows when to request confirmation. Sometimes they consult notes stored outside the official system. These activities may look like