Max Paradox

PROBLEM? RELAX. YOU'VE GOT THIS!

A tongue-in-cheek guide series ;)

Every Project Has a New Team, and Every Time We Start From Scratch

How to Quickly Define Roles, Ways of Working, and Communication Rules in Project Teams That End Before Everyone Has Learned Each Other's Names



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

Every Project Has a New Team, and Every Time We Start From Scratch

How to Quickly Define Roles, Ways of Working, and Communication Rules in Project Teams That End Before Everyone Has Learned Each Other's Names

Max Paradox

PROBLEM? RELAX. YOU'VE GOT THIS!

ISBN: 978-83-8483-674-3

CONTENTS

INTRO 5

Chapter 1 - Why Experienced People Keep Starting From Scratch 12

Chapter 2 - The Most Expensive Word on a Project Call: "Obviously" 22

Chapter 3 - Everyone Is Responsible, So Nobody Knows Who Owns the Result 32

Chapter 4 - The Magnificent Kickoff Meeting That Accomplishes Absolutely Nothing 41

Chapter 5 - Five People Assigned, One Person's Worth of Time 51

Chapter 6 - The Cost of Being Agreeable When Someone Needs to Speak Up 61

Chapter 7 - Everyone Finished Their Part, but the Project Still Doesn't Work 72

Chapter 8 - Three Chat Platforms, Seven Threads, and No Current Decision 82

Chapter 9 - The Team Working Agreement: A Few Rules That Prevent a Thousand Arguments 93

Chapter 10 - Week by Week, Without Turning Progress Reports Into Performance Art 103

Chapter 11 - The Decision Is Needed on Tuesday, but the Committee Meets on Friday 113

Chapter 12 - "While We're at It": How a Small Project Becomes an Unfunded Empire 123

Chapter 13 - A New Person Joins Halfway Through: How to Get Them Productive Without Starting the Project Again 134

Chapter 14 - Two Departments, Two Priorities, and One Project Stuck in the Middle 144

Chapter 15 - When the Project Falls Apart: Rescue the Work Before Organizing the Blame 154

Chapter 16 - When One Person Keeps Missing Deadlines and Everyone Else Starts Walking on Eggshells 164

Chapter 17 - A Team Without a Shared Boss: How to Get Results When Nobody Reports to You 175

Chapter 18 - When Chaos Quietly Returns: Recognizing the Early Signs Before the Team Unravels 185

Chapter 19 - The Project Is Finished. Who Owns Tuesday Morning? 195

Chapter 20 - The Next Project Has a New Team. You Don't Have to Start From Zero 204

INTRO

At 4:17 on a Thursday afternoon, someone asks a perfectly reasonable question in the project group chat: "Who is actually approving this?" Seven people have seen the message. Three have reacted with a thumbs-up, which is an impressive achievement considering that nobody has answered the question. One person suggests checking with Finance. Another believes Operations already signed off. The project manager remembers a conversation with someone from Legal, although that person left the project last Tuesday. Meanwhile, the supplier is waiting for confirmation, the deadline is tomorrow, and the entire team has suddenly developed a remarkable interest in the precise meaning of the word "approval." Welcome to temporary project teamwork, where everyone has impressive qualifications and nobody can confidently identify the person holding the metaphorical pen.

The frustrating part is that this probably isn't your first project. It might not even be your twentieth. You've worked with talented people, survived impossible deadlines, navigated organizational politics, and attended enough kickoff meetings to recognize the opening slides before they're shared. Yet every new assignment seems to come with the same introductory challenges. Who makes decisions? How quickly should people respond? Where are the latest documents? What happens when two departments disagree? And why does the person who knows the answer always appear to be on vacation? You would think that experience should make these questions disappear. Instead, organizations develop an extraordinary ability to forget how teamwork works every time they rearrange the participants.

There is a reason for this. Experience belongs to individuals, while effective collaboration belongs to relationships and working arrangements. You can assemble six outstanding specialists and still create a remarkably ineffective team if their assumptions about cooperation are incompatible. One person expects immediate decisions. Another believes every important choice requires consultation. Someone prefers written instructions, while somebody else considers an informal conversation sufficient authorization to proceed. Each person may be behaving sensibly according to the practices that worked in previous assignments. Unfortunately, a collection of sensible individual habits doesn't automatically produce a sensible shared system. Sometimes it produces an organizational traffic intersection where everyone has the right of way and nobody has thought to install traffic lights.

Traditional responses don't always help. Organizations launch increasingly elaborate onboarding programs, introduce new project management software, distribute responsibility matrices, and organize workshops with enthusiastic names involving alignment, transformation, or collaborative excellence. Some of these tools are genuinely useful. Others merely give confusion a more attractive interface. A beautifully formatted project charter cannot make an unavailable specialist available. A shared dashboard cannot resolve a disagreement over decision-making authority. And a motivational presentation about teamwork won't change the fact that three people have been assigned the same deliverable while another essential task has no owner whatsoever. Problems rooted in unclear expectations need actual agreements, not simply more places to record that everyone is committed to solving them.

Temporary teams face an additional challenge: they have almost no time to become comfortable with one another. In a permanent department, people gradually learn who needs detailed explanations, who spots potential problems early, and who says "I'll get back to you shortly" with a personal definition of shortly that includes the next financial year. Short projects cannot afford that lengthy discovery process. By the time everyone understands how their colleagues operate, the assignment may already be approaching its final review. The practical answer is to make certain expectations explicit earlier, without attempting to replace human judgment with a hundred-page rulebook. A small amount of deliberate coordination can prevent an enormous amount of accidental confusion.

That doesn't mean turning every project into a miniature bureaucracy. You shouldn't need a governance committee to decide whether two colleagues can exchange information. Nor should a five-week assignment require a documentation system more complicated than the product being developed. The challenge is learning which agreements genuinely matter and which formalities merely create additional work. Some projects need clear approval boundaries. Others depend primarily on reliable handovers between specialists. Still others suffer because people are assigned work without the time or authority to complete it. The appropriate solution depends on the obstacle. A useful rule should make cooperation easier, not become another obligation that people must somehow fit around their actual responsibilities.

Throughout this book, we'll explore how to recognize those obstacles before they become expensive surprises. You'll learn how to distinguish responsibility from authority, clarify what a completed task really means, and recognize

when apparent progress hides unfinished dependencies. We'll examine communication arrangements that help people find current decisions without conducting digital archaeology through months of messages. We'll also look at practical ways to handle competing priorities, reluctant decision-makers, unexpected changes, difficult conversations, and colleagues who contribute to five different projects while apparently possessing only one physical body. The objective is not to eliminate every disagreement or interruption. It is to make the next useful action clear enough that people can take it.

Consider what happens when a team starts work without checking its first critical dependency. A marketing specialist assumes the technical team will provide final product specifications. The technical team assumes marketing has already confirmed the requirements. Procurement prepares an order based on an earlier draft. Nobody is deliberately obstructing anybody else, but each person is waiting for a different signal. A brief conversation at the beginning might have exposed the mismatch. Later, the same clarification requires rework, revised commitments, and an uncomfortable explanation to the sponsor. The difference isn't intelligence or dedication. It is whether the team recognizes that the space between individual responsibilities deserves as much attention as the responsibilities themselves.

Of course, good coordination cannot magically solve structural problems. If the organization assigns the same specialist to four urgent initiatives, no communication technique will manufacture additional working hours. If a sponsor insists on an impossible deadline, carefully worded meeting notes won't make that deadline achievable. And if a

supplier is contractually unable to deliver something, a cheerful attitude won't rewrite the agreement. Practical teamwork means recognizing these limits, making their consequences visible, and identifying who can authorize a change. Sometimes the right decision is to reduce scope, adjust expectations, or admit that a commitment cannot be met. That may be less exciting than promising to deliver everything through extraordinary dedication, but it is considerably more useful to the people who must live with the outcome.

You also don't need to be the official project manager to benefit from these ideas. Perhaps you're a specialist joining a cross-functional initiative, a coordinator managing one important workstream, or a department manager lending people to a temporary team. You may have limited authority over the other participants and no opportunity to redesign the organization's procedures. You can still make a difference by asking what information a colleague needs before starting, clarifying which decision remains unresolved, or making the consequences of a missed deadline visible early enough for someone to act. Effective collaboration often improves through small, timely interventions rather than grand organizational announcements. Nobody needs to appoint you Supreme Director of Common Sense before you can ask a useful question.

There will also be failures. People will misunderstand instructions, underestimate their workload, forget agreements, and occasionally produce an elaborate explanation of why the task they promised to finish was never technically their responsibility. Some difficulties will be genuinely complicated. Others will be absurd in ways that

deserve a moment of appreciation before you attempt to fix them. Humor helps us notice the ridiculous patterns that emerge when perfectly ordinary people work inside imperfect systems. It shouldn't be used to dismiss genuine exhaustion, unfair expectations, or serious mistakes. But if you've ever attended a meeting to decide whether another meeting is necessary, you've already encountered material that would be difficult for a professional comedy writer to improve.

The real goal is straightforward: each new team should be able to become useful faster, with less wasted effort and fewer avoidable misunderstandings. You won't know every personality, predict every obstacle, or construct a flawless plan. You can, however, learn to establish enough clarity that people understand what they're trying to achieve, how their contributions fit together, and what to do when reality refuses to follow the presentation. Those abilities travel with you from project to project. They don't disappear when the team is dissolved, the shared channel is archived, or somebody creates another folder called "Final Version - Updated Final."

And when the next invitation arrives, with another unfamiliar list of attendees and another ambitious delivery date, you won't need to pretend that everyone already knows how to work together. You can help them figure it out before the expensive misunderstandings begin. You may still struggle to remember everyone's name, especially if three people are called Alex and two have identical profile photographs. That's manageable. A functioning team doesn't require perfect name recognition by Friday. It requires something far more valuable: enough shared

understanding to get meaningful work done while everyone is still learning who everyone else is.

Chapter 1 - Why Experienced People Keep Starting From Scratch

Somewhere in your organization, there is a person who knows exactly why the last customer onboarding project ran three weeks late. Unfortunately, nobody knows who that person is. The project manager has moved to another department, the technical lead has joined a different company, and the remaining documentation consists of a cheerful completion report describing the implementation as "a valuable learning experience." The report contains several photographs of smiling employees and an impressive diagram of the original schedule. What it does not contain is an explanation of why the integration supplier needed an additional approval, which nobody discovered until installation was supposed to begin. Six months later, a new team launches a remarkably similar initiative. Everyone is experienced, enthusiastic, and determined to avoid previous mistakes. They promptly make the same mistake, although this time with a different supplier and considerably more sophisticated reporting software.

This is one of the most expensive peculiarities of project-based organizations. A company can complete dozens of initiatives, employ highly qualified specialists, and invest heavily in professional development while continuing to behave as though every new assignment represents humanity's first attempt at coordinated work. Individuals accumulate experience, but the organization struggles to make that experience available to the next group. Knowledge travels with people, and people travel elsewhere. Processes change, project names change, and the lessons

supposedly learned during completion meetings quietly disappear into storage systems designed primarily to demonstrate that storage systems exist.

The underlying difficulty is that experience has several different forms. A financial analyst may understand the budgeting process extremely well without knowing how a particular marketing department handles late changes. An engineer may recognize a technical risk immediately but have no idea which executive can authorize the necessary response. A project manager may have delivered twenty successful implementations using arrangements that are completely unsuitable for the current team. Each participant brings useful knowledge, but not necessarily knowledge of how these particular people must cooperate under these particular constraints. The new team therefore possesses considerable professional ability while lacking the shared understanding needed to use it effectively.

Expertise doesn't automatically become teamwork

Imagine a company preparing to introduce a digital visitor management system across its office buildings. The project requires facilities specialists, IT professionals, security representatives, reception managers, and an external technology provider. Everyone has worked on comparable initiatives. During the first planning meeting, each person presents a sensible approach based on previous experience. Facilities wants the installation schedule finalized before ordering equipment. IT prefers completing technical validation before committing to installation dates. Security requires approval of access arrangements, while reception managers want to review the visitor journey before anything becomes permanent. The supplier needs confirmed quantities before reserving its installation team. Nobody is

being unreasonable. Unfortunately, every sensible requirement depends on somebody else's equally sensible requirement.

An inexperienced observer might conclude that these specialists need better training in cooperation. The more useful conclusion is that they have not yet established the relationships between their individual responsibilities. They know how to perform their jobs, but they do not know which parts of their work must happen together, which decisions are already settled, or which assumptions remain untested. A room full of experienced people is not automatically an experienced team. The distinction matters because additional expertise may accomplish very little when the obstacle is the arrangement connecting that expertise.

Permanent teams gradually develop a shared operating history. They learn which details another department needs before accepting a deliverable, how much notice particular decisions require, and which apparently routine requests involve hidden complications. Much of this understanding is never formally recorded because everybody involved already knows it. When the team dissolves, those assumptions remain distributed across people's memories. The next team may receive the final deliverables but not the reasoning, relationships, and practical knowledge that made them possible. It's rather like inheriting a fully equipped workshop without knowing which machine occasionally overheats, which switch controls the ventilation, or why everybody avoids the cabinet marked "miscellaneous."

The obvious response is to document everything. Unfortunately, everything is an ambitious quantity of information. Organizations that attempt comprehensive knowledge preservation often produce enormous

repositories containing outdated instructions, conflicting versions, and detailed accounts of decisions that no longer matter. Finding the useful material becomes a separate research assignment. New project members may receive access to hundreds of documents and be told that all the information is available. Technically, this may be true. Practically, it resembles telling someone that the answer to their question is somewhere on the internet. Access and usefulness are not interchangeable.

What should actually survive a project?

The most valuable knowledge is usually connected to decisions, constraints, dependencies, and mistakes that could reasonably recur. In the visitor management example, the next team would benefit from knowing that security approval must precede a particular installation step, that one building requires different equipment, and that the supplier's lead time begins only after formal confirmation of specifications. These facts influence planning. A detailed record of every discussion about the color of the reception screen probably does not, unless that color somehow caused a significant problem. The purpose of preserving knowledge is to improve future decisions, not to establish a museum dedicated to previous meetings.

A useful project record should therefore distinguish between what happened and why it matters. "Installation started late" describes an outcome. "Installation started late because building access permissions were requested only after the contractor arrived" identifies a cause that another team can address. Better still, the record might state when access permissions need to be requested and who is authorized to approve them. This transforms experience into something operational. The next project manager does not

merely inherit a story about an unfortunate Tuesday. They inherit a practical consideration that can influence the schedule before the first contractor arrives.

There is, however, a danger in turning previous experience into unquestionable procedure. A solution that worked last year may no longer fit the current circumstances. Perhaps the supplier has changed its process, the organization has different requirements, or new technology has removed an earlier limitation. Historical knowledge should help the team identify questions worth checking. It should not automatically dictate the answers. If an old project required ten days for approval, the new team should verify whether that condition remains valid rather than treating the number as an ancient organizational commandment carved into the procurement department's furniture.

One practical way to use previous experience is to conduct a short investigation before detailed planning. Find out whether the organization has completed a comparable initiative and identify somebody who understands its important difficulties. Instead of requesting every document, ask which assumptions proved wrong, which approvals took longer than expected, and which activities depended on information that arrived too late. Ask what that person would verify first if they had to repeat the project. A focused conversation can reveal the kind of knowledge that rarely survives in formal documentation, particularly the small operational details that determine whether a schedule is realistic.

In our visitor management project, a former facilities coordinator might explain that the technology installation was straightforward but access to occupied reception areas

had to be negotiated separately with each building manager. The new team originally planned installations according to supplier availability alone. That one observation changes its approach: building access becomes an explicit prerequisite rather than a minor administrative detail. The coordinator has not provided a miraculous management framework. They have prevented the team from purchasing another expensive opportunity to learn exactly the same lesson.

The difference between knowing and assuming

Another reason teams repeatedly start from scratch is that they fail to distinguish established facts from familiar assumptions. A manager says the procurement process normally takes a week. An experienced colleague believes the supplier can deliver by the end of the month. Someone remembers that the relevant department doesn't require additional approval for this type of expenditure. These statements may all be reasonable starting points. They become dangerous when transferred into the project schedule as confirmed information without verification.

The distinction is easiest to maintain when the team openly separates three categories: what has been confirmed, what is currently assumed, and what remains unknown. This need not involve a sophisticated database. A short working note can identify an important assumption, its potential consequence, and the person who will check it. For example, if installation depends on permission to work outside normal office hours, someone should confirm that permission before the schedule becomes a promise. If the assumption turns out to be correct, excellent. If it is wrong, the team discovers the problem while alternatives still exist.

The important word here is important. Nobody needs to investigate every conceivable uncertainty before starting

work. A short project would never begin if participants insisted on complete knowledge of the future. Concentrate instead on assumptions that could materially affect cost, timing, quality, or the feasibility of the intended result. If a minor detail can be corrected easily later, the team may reasonably proceed. If an unverified condition could invalidate several weeks of preparation, checking it early is worth the effort. Good planning does not require omniscience. It requires knowing which unknowns deserve attention.

This approach also changes the atmosphere of initial conversations. When participants are encouraged to distinguish assumptions from facts, admitting uncertainty becomes an ordinary part of responsible planning. People no longer need to present every estimate with artificial confidence. A specialist can say, "I believe this is possible, but I need confirmation from the supplier by Thursday." That statement is much more useful than a cheerful assurance followed by an increasingly elaborate explanation when Thursday's deadline arrives. The first gives the team something to manage. The second offers optimism as a substitute for information.

Make knowledge portable without making it enormous

Temporary teams benefit from a small amount of information that allows participants to become productive quickly. A newcomer should be able to understand the purpose of the project, the current state of work, the major constraints, and the decisions that remain open. They should also know where authoritative information is kept and whom to contact about important dependencies. This is different from giving everybody a complete record of everything the organization has ever discussed. A well-prepared starting

point reduces the amount of history each person must reconstruct before contributing.

Consider someone joining the visitor management initiative halfway through. They do not need to review every rejected interface design or listen to recordings of six earlier workshops. They need to know which solution was selected, which buildings are included, what remains incomplete, and which commitments must be honored. Historical explanations become relevant where they illuminate a current restriction or prevent repetition of a previous mistake. By introducing information in that order, the team respects the newcomer's time without pretending that previous decisions were made for no reason.

The responsibility for making knowledge portable cannot rest entirely with a departing project manager. If the organization repeatedly relies on one exceptionally experienced individual to remember every informal arrangement, it has created a vulnerability disguised as a valuable employee. That person may be enormously helpful, but their absence should not disable ordinary operations. Critical knowledge needs to become accessible to the people who depend on it. This doesn't mean requiring specialists to document every thought. It means identifying information whose disappearance would force others to repeat expensive work or make avoidable mistakes.

A similarly useful habit is capturing lessons when they are still fresh. Waiting several months after completion makes it harder to remember the precise sequence of events and the conditions behind decisions. A short discussion near the end of the project can identify the two or three things a future team should examine before attempting something similar. The result should be specific enough to influence an actual

decision. "Improve stakeholder engagement" is a respectable aspiration, but it won't tell the next manager that regional office administrators require ten working days to arrange building access.

There will still be situations where previous knowledge is incomplete, unavailable, or too expensive to recover. Perhaps the experienced employees have left, the organization has changed, or the project is genuinely unlike anything attempted before. In those cases, the sensible response is to recognize the uncertainty and investigate the most consequential unknowns. Pretending that institutional experience exists when nobody can locate it is unlikely to improve the outcome. A team can work effectively with incomplete information provided it understands where the gaps are and avoids treating guesses as settled agreements.

Repeatedly starting from scratch is therefore neither an unavoidable feature of temporary teamwork nor a problem solved by collecting more documents. The real opportunity is to preserve and transfer the information that helps people make better decisions together. A new team will always need to establish relationships and understand its particular assignment. It should not also have to rediscover why a known constraint matters, repeat a failed experiment, or spend its first week searching for an answer the organization already paid somebody to find.

The next time you join a project, experience may not arrive in a convenient package marked "Everything You Need to Know." More likely, it will be scattered across people, files, and imperfect memories. Your advantage is knowing what to look for. Identify the important decisions, investigate consequential assumptions, and recover the practical lessons that remain relevant. You won't eliminate

the unfamiliarity of a new team, but you can prevent unfamiliarity from becoming unnecessary ignorance.

And if somebody proudly announces that the company has completed this exact kind of project fifteen times, there is one perfectly reasonable question worth asking: "Wonderful. What did we learn?" The length of the resulting silence may tell you more about the organization's project management maturity than its entire collection of completion certificates.

Chapter 2 - The Most Expensive Word on a Project Call: "Obviously"

"Obviously, we'll need a final review before launch." It's a sentence capable of receiving immediate agreement from everyone in a meeting while communicating almost no useful information. To the product manager, final review means confirming that the proposed features meet the business requirements. To the quality specialist, it means completing a defined set of tests. To the communications team, it means checking that public-facing materials are accurate. To the executive sponsor, it means a brief demonstration followed by a decision to proceed. All four people believe they have agreed on a sensible requirement. They have actually agreed that something called a final review should happen, while imagining four different events. When launch week arrives, each wonders why everyone else seems so confused about something that was perfectly obvious.

The danger of familiar language is that it discourages questions. Nobody wants to interrupt a professional discussion to ask what "ready" means. Everybody has been using that word since childhood. Likewise, asking what counts as approval can feel unnecessarily pedantic when several senior colleagues have just confidently agreed that approval is required. Yet shared vocabulary does not guarantee shared expectations. Words acquire operational meaning through experience, and people who have worked in different environments often bring different definitions to the same conversation. The disagreement remains invisible until somebody acts on an interpretation that another person never intended.