

The PARIS Method

Part 1: The PARIS Method

PARIS is a simple five-part structure for thinking clearly about a challenge — where it came from, what to do about it, what success looks like, what it teaches you, and what it takes to pull off.

You may already know a version of it. STAR (Situation, Task, Action, Result) is the classic interview-prep tool, and PARIS builds on the same foundation. But PARIS adds two things that make it more than an interview technique: it asks you to name what's actually at *risk*, not just describe the situation neutrally — and it closes the loop with **Insights** and **Skills**, turning a story about what happened into material you can actually learn from and grow with.

That last part matters most. A good PARIS isn't just well-told — it leaves you (or whoever you're helping) a little more capable than before you told it.

The five elements

P — Problem What's the problem? Not just "what happened," but what was actually going on.

A well-framed Problem has three parts:

- **Situation** — a neutral description of the context. What's the setting? Who's involved? What was supposed to happen?
- **Complication / challenge** — what made this hard? What was uncertain, difficult, or new?
- **Risk** — what was at stake? What might have happened if nothing was done, or if it went wrong?

This last part — Risk — is easy to skip, and it's the one most worth slowing down for. Without it, a Problem reads as a task ("we needed to launch a product"). With it, a Problem reads as a reason anyone should care ("we needed to launch before a competitor did, in a market we'd never operated in, on a timeline that left no room for mistakes"). The size of the Problem — including the risk inside it — is what gives the eventual Result its weight. A small problem, honestly told, produces a small story. A real problem, honestly told, produces a real one.

A — Action What needs to be done? Or, afterward: what did you actually do?

Keep this part leaner than you think you need to. If the Problem is well-defined, the Action mostly follows logically from it — and a long list of everything you did is usually less convincing than a short, clear account of the choices that mattered.

R — Result What needs to be achieved? Or: what actually happened?

Make it as measurable and concrete as you honestly can. A result that can be counted, compared, or observed is more useful — to you, and to anyone you're telling it to — than one that can only be felt.

I — Insights What can we learn from this? What did you learn? Is there a different way to look at the problem itself?

This is where PARIS earns its place beyond a status update. Insights isn't just "and here's a lesson I'll tuck away" — it's the moment you step back and ask whether you're even framing the problem the way it deserves to be framed. Sometimes the real insight isn't about what you did, but about what you thought the problem was in the first place.

S — Skills What skills or strengths were used — or built — along the way? What skills would this take, and do you already have them, or do you need to develop or borrow them?

Skills can point in two directions. Looking back, it's evidence: here's proof of what I can do. Looking forward, it's a gap check: here's what this will take, and here's what I'm missing — which might mean learning something, or simply asking for help.

Not a line — a loop

It's tempting to treat PARIS as a form: fill in P, then A, then R, then I, then S, in order, once. In practice, it rarely works that way — and it shouldn't.

The most useful moment in PARIS is often when Result forces you back to Problem. You sketch out what success would look like, and it makes you realize the Problem you named wasn't quite the real one. Or you start listing Actions, and one of them exposes a Risk you hadn't named yet. Each element tests the others. Problem and Result, in particular, function as two anchors that keep checking each other: does this Result actually resolve that Problem? Is this Problem big enough to justify caring about this Result?

So rather than a straight line —

P → A → R → I → S

— think of PARIS more like a loop you circle through until the picture is clear:

P ⇌ A ⇌ R, tested and retested, with **Insights** surfacing along the way rather than only at the end, and **Skills** clarifying what all of this will actually take.

This matters more the further you move from "telling a story about the past" and toward "using PARIS to think something through." A finished story has a fixed shape — the Problem happened, the Result happened, and PARIS mostly just organizes the telling. A problem you're still inside of doesn't have a fixed shape yet. You may need to revisit your own PARIS more than once — even repeatedly, over the life of a project — as you learn more, as circumstances shift, and as your sense of what's actually at stake evolves.

Two directions, one method

Everything above works the same in two directions:

Backward — you're making sense of something that already happened. This is PARIS as reflection: as a way to prepare a story for an interview or a performance review, as a way to debrief a project, as a way to help someone (or yourself) notice what actually happened underneath the version they're used to telling.

Forward — you're using PARIS before or during something, to think it through. This is PARIS as a planning and self-coaching tool: naming the real Problem before diving into Action, deciding upfront what Result would actually mean, checking honestly what Skills the work will take.

The five elements don't change between these two directions. What changes is the tense — and, sometimes, the question underneath the question. Looking back, Insights asks "what did this teach me?" Looking forward, it asks "what am I assuming, and what might I be missing?" Looking back, Skills asks "what did I demonstrate?" Looking forward, it asks "what do I still need?"

Individual, or shared

PARIS also works whether you're doing it alone or with others.

Used alone, it's a way to think — a structure for a journal entry, a moment of reflection before an interview, a five-minute gut-check before starting something hard.

Used with others, it becomes a shared reference: a coach helping a client see their own story more clearly, a manager and a report aligning on what a project is really about before work begins, a whole team defining — and later revisiting — what they're actually trying to achieve together.

That's where PARIS gets genuinely useful for organizations, not just individuals: a team's PARIS isn't fixed the day the project starts. It's worth returning to. Has the Problem changed? Is the Result you agreed on three weeks ago still the right one? A project's PARIS is less like a form you fill in once, and more like a living reference you keep checking yourself against.

The next part walks through what this looks like in practice — the same situation, seen through five different roles: someone telling their story after the fact, a coach helping a client discover what they haven't yet said, someone using PARIS to think clearly before taking something on, a manager coaching a team member through it, and a team defining — and redefining — their PARIS together as a project unfolds.

One Situation, Five Views

Before we walk through how PARIS shows up in different roles, let's set the scene. Throughout this booklet, we'll return to the same situation — seen from a different angle each time. Watch one story unfold through five lenses to see what changes, and what stays the same, as we move from backward-looking to forward-looking, and from individual to team.

Here's the situation.

Inkwell makes a touch-screen e-reader with a well-loved interface — smart search, predictive text for notes and annotations, a clean touch-and-swipe navigation system. It's been a genuine success across Germany and the rest of Europe. Now leadership wants it launched in Taiwan, and fast: eight weeks, driven by the fear that a competitor could get there first. Taiwan is also meant to be a test case — if it goes well, the same playbook will be used to launch across the rest of East Asia.

On paper, it looks straightforward. The hardware works. The software works. Why not just translate the interface and ship it?

In reality, three things complicate the picture:

- **The interface doesn't translate as cleanly as the words do.** Predictive text and search were built around Latin-alphabet input — word boundaries, autocomplete, even the logic of a keyboard. None of that maps cleanly onto Traditional Chinese. Translating the labels was the easy part; the input experience underneath still behaves like it's from somewhere else.
- **A regulatory requirement** in Taiwan — around data handling — wasn't part of the original design. It's solvable, but it wasn't budgeted for, and nobody's quite sure yet how much it will cost in time.
- **The engineering team that owns the core product sits in a different timezone,** several hours removed from the local launch team. Every adjustment, however small, now comes with a communication delay.

Mei is the product manager responsible for making this happen. She's based in Taipei, she's excited about the opportunity, and she's never led a launch quite like this before. Her manager, **David**, wants her to succeed — but he also knows this is the kind of project where the biggest risk isn't the obvious one on the surface. Around Mei sits a small cross-functional team: local operations, a designer, and an engineering lead dialing in from Inkwell's headquarters in Germany.

Eight weeks. A product that works. An interface that doesn't quite work the same way in Taiwan as it did back home.

What follows is how PARIS plays out for each of them — Mei, telling the story afterward; a coach, helping Mei see what she's not yet saying; Mei again, using PARIS *before* the sprint even starts; David, coaching her through it; and the team, defining and redefining their PARIS together as the eight weeks unfold.

Same situation. Five very different conversations.

Part 2: PARIS in Practice

Chapter 1 — Telling Your Story (Job Seekers & Professionals)

This is PARIS in its most familiar form: looking back at something you've already done, and shaping it into a story you can tell — in an interview, on a resume, or simply to yourself, as a way of taking stock of what you've accomplished.

It's backward-looking, and it's self-directed: you're the one deciding what the story is, even if you're eventually telling it to someone else.

From experience to story

A year after the Taiwan launch, Mei is interviewing for a new role. The interviewer asks: "Tell me about a time you led something you'd never done before."

(The same story works just as well in a performance review, when Mei's manager asks her to reflect on the year — the shape doesn't change, only the audience and the purpose.)

Mei could answer in a sentence: "I led our first launch in Taiwan, and it went well." True, but it tells the interviewer almost nothing about how Mei thinks, what she's capable of, or what she'd bring to a new team. PARIS is how she turns that sentence into an actual story.

Problem Mei starts with the situation, not with herself: "Inkwell had built a well-loved e-reader in Germany, and leadership wanted to launch it in Taiwan within eight weeks — partly to beat a competitor to market, and partly as a test case for the rest of East Asia." She names the complication honestly: "The product worked, but the input experience — predictive text, search — had never been built for Traditional Chinese, and nobody had budgeted time for the data-handling requirement Taiwan's regulations would require." And she names what was at stake: "If Taiwan went badly, it wouldn't just be one disappointing launch — it would undermine the company's confidence to expand into the region at all."

Action Mei keeps this part tight. "The moment I understood the scope of the input problem, I reprioritized the roadmap around it. I brought in our Germany-based engineering lead early instead of late, and I pushed for a local legal review to run in parallel with development rather than after it — that's the one decision I think saved us the most time."

Result "We launched on schedule. Our early adoption numbers matched the German launch's first month — a bar nobody expected us to hit. And because the regulatory review ran in parallel, we caught the data-handling gap in week three instead of week seven."

Insights Here's where Mei's story becomes more than a summary of what happened. "Going in, I'd assumed 'translation' meant language — and it took a frustrated week of user testing to realize the real problem was input, not vocabulary. If I took on something like this again, I'd ask 'what does this product actually depend on, underneath the language it's written in?' on day one, not week two."

Skills She names what the experience built, not just what she already had. "It taught me how to coordinate across timezones with a team I'd never worked with before, and gave me the

confidence to run technical and regulatory work in parallel under pressure, instead of waiting for a 'clean' sequence."

For the resume: Result + Action only

A resume doesn't have room for the full story — and doesn't need it. There, Mei uses only Result and Action, in the form: **Achieved (result) by doing (action)**.

Delivered Inkwell's first Taiwan launch on an 8-week timeline, matching first-month adoption in the home market, by identifying an overlooked input-localization gap early and running technical and regulatory work in parallel.

For the interview: PAR(IS)

In conversation, the order matters differently than on paper. Mei always starts with Problem — never Action — because a story that opens with "I reprioritized the roadmap" gives the interviewer nothing to measure it against. Only once the Problem (and its risk) is on the table does the Action mean anything.

She then moves through Result, and brings in Insights or Skills depending on what the interviewer is actually asking. If they've asked "tell me about a challenge," she leans on Insights. If they've asked "what are your strengths," she leans on Skills. The full PARIS is there if needed — but she doesn't recite all five elements every time.

Try it

Think of something you did that you're quietly proud of — not necessarily dramatic, just real. Write down the Problem first, and specifically its Risk: what would have happened if nothing had been done, or if it had gone wrong? Notice how much of the story that one sentence is already doing.

Chapter 2 — Discovering the Story (Coaches)

Chapter 1 showed Mei's story once it was already well-shaped. But stories don't start out that way. Most people, asked to describe something they did, tell it flatly — underselling the Problem, skipping the Risk, and rushing past the Insight without noticing there was one to find.

This is where a coach's job differs from an interview coach handing someone a template. You're not teaching Mei the PARIS structure so she can fill it in correctly. You're using it to notice what she's leaving out — and helping her notice it too.

This chapter is still backward-looking, but it's no longer self-directed. It's Mei's story, and someone else's questions.

Before the story was a story

Some months after the Taiwan launch — before she's ever told it in an interview — Mei is in a coaching session, and her coach asks her to describe something she's proud of from the past year. Here's what Mei actually says, in her own words, before any shaping:

"We launched in Taiwan. It was a tight timeline, but we made it work. The team pulled together and we got it done."

That's it. Four sentences. If the coach took this at face value, there'd be nothing to work with — no real Problem, no Risk, no Action worth naming, and definitely no Insight. But the coach's job isn't to accept the story as told. It's to notice what's missing, and ask.

What the coach listens for

On Problem — "You said 'tight timeline.' What made it tight? What was actually at stake if you'd missed it?"

Mei's first answer is practical: "We wanted to beat a competitor." The coach doesn't move on. "And if you hadn't?" Only with a second and third question does Mei arrive at something bigger than she'd offered on her own: that Taiwan wasn't just one launch — it was leadership's test case for the whole region, and a stumble there might have meant no East Asia expansion at all. That's a materially bigger Problem than "tight timeline" — and Mei hadn't led with it, not because she was hiding it, but because it hadn't occurred to her that it mattered to the story.

On Action — "You said 'the team pulled together.' What did you actually do?"

Mei defaults to "we," the way many people do when a project succeeded — it feels more honest, more modest. The coach isn't trying to inflate her ego; the goal is accuracy, not flattery. Gently pressed, Mei separates out her own specific choices: reprioritizing the roadmap, pulling engineering in early, running legal review in parallel. Small distinction, but it changes the story from something that happened around her to something she did.

On Insight — "What did you learn about how you'd been thinking about the problem?"

This is the question Mei can't answer quickly, and it's usually the most valuable one. Her first instinct is a generic answer: "I learned that cross-team coordination is important." True, but not

really an insight — more a platitude anyone could offer without having lived through the project.

The coach stays with it: "Was there a moment where you realized the problem wasn't what you thought it was?" And here Mei pauses, and finds it — the week where user testing revealed that "localizing" the product meant nothing without fixing input, not just translation. She'd half-forgotten this, because by the time the project succeeded, it just felt like part of the process. The coach's role wasn't to hand her that insight. It was to keep asking until she found it herself.

Discovery, not extraction

Notice what didn't happen in this session: the coach never told Mei what her Problem, Insight, or Skills were. Everything above came from Mei — the coach's job was only to keep asking, especially in the places where Mei's own account was thinnest.

A useful signal for a coach: **whichever letter the client rushes past fastest is usually the one worth slowing down on.** People tend to under-describe the Problem when they're modest, under-describe the Action when they default to "we," and skip Insight almost entirely when the experience hasn't yet been turned into language. None of that is dishonesty — it's just what happens when a story hasn't been asked about yet.

Try it

Next time someone tells you a flat, four-sentence version of something they did, resist the urge to move on. Pick the element that felt thinnest — usually Problem's Risk, or Insight — and ask one more question about it before you respond. Notice what changes in what they tell you next.

Chapter 3 — Thinking It Through Before You Start (Self-Coaching)

The first two chapters looked backward — a story already lived, told once well and once more honestly. This chapter turns PARIS around. Mei hasn't launched anything yet. It's week zero, and she's just been told the Taiwan launch is hers.

This is PARIS as a pre-mortem, not a debrief — and it's still self-directed. Nobody is asking Mei these questions. She's asking them of herself, before Action has even started, precisely so she doesn't have to discover the real Problem three weeks in.

Before touching the roadmap

Mei's instinct, like most people's, is to start with Action: open a project plan, list tasks, assign owners. PARIS asks her to resist that instinct and sit with Problem first — not because planning doesn't matter, but because a plan built on the wrong Problem is a plan for the wrong thing.

Problem — what am I actually solving?

Mei's first draft answer is the one she's been given: "Launch the e-reader in Taiwan in eight weeks." That's a task, not a Problem. She pushes further, the way her future coach will later push her to: What's the situation, really? What's the complication? What's actually at risk?

Written out, it looks more like: *"Inkwell needs to prove it can succeed outside its home market — and Taiwan is the test. The product works at home, but I don't yet know what breaks when it meets a different language, different regulations, and a team I've never worked with. If this goes badly, it's not just a missed launch — it could cost the whole East Asia expansion."*

That's a very different starting point than "launch in eight weeks." It's bigger, more honest about what's unknown, and — importantly — it's a Problem Mei can actually test her plan against later.

Action — not yet. First, Result.

Before deciding what to do, Mei asks what she's actually aiming for — because "launch on time" isn't specific enough to plan against. She writes down what a good Result would look like: on schedule, adoption in the first month close to the German launch's numbers, and no regulatory surprises discovered after launch rather than before it. Only with that in view does Action become a real question rather than a guess: what needs to happen for that Result to be realistic in eight weeks?

Skills — what will this take, and what am I missing?

This is the part self-coaching makes easiest to skip, because admitting a gap can feel like admitting weakness before you've even started. Mei asks herself plainly: what does this actually require? Fluency in navigating regulatory processes — she has some, not enough. A real working relationship with the Germany-based engineering team — she doesn't have one yet. She can't manufacture experience in a week, but she can decide, now, to build the engineering relationship early and to bring in legal counsel sooner than instinct would suggest. Knowing what she's missing changes what she does with week one.

Insight — what am I assuming?

Looking back, Insight asks "what did I learn?" Looking forward, it asks something else: "what am I assuming right now that might not hold?" Mei writes down her biggest assumption — that "localizing for Taiwan" mostly means translation. She doesn't yet know this assumption is wrong. But naming it, in writing, before she's committed to a plan built on it, means she's more likely to notice early if reality disagrees. (It will. But because she asked the question in week zero instead of never, she's positioned to notice it in week two rather than week seven.)

The value isn't the answer — it's asking early

Notice that Mei's week-zero Problem statement isn't fully correct. She doesn't yet know that input, not translation, is the real localization issue — that discovery is still ahead of her.

Self-coaching with PARIS doesn't guarantee you'll see everything coming. What it does is put you in the habit of asking the right questions before you're deep enough into Action that a wrong assumption becomes expensive to unwind.

This is also where the loop from Part 1 shows up most clearly at the individual level: Mei's Problem, Result, and Skills statements from week zero aren't fixed. They're a first draft — one she can, and should, come back to.

Try it

Before your next project, meeting, or difficult conversation, write down your Problem — situation, complication, and risk — in a few sentences, before you plan anything. Then write down what Result you're actually aiming for. Notice whether the Action you'd instinctively jump to still makes sense once both are on paper.

Chapter 4 — Coaching Before the Work Begins (Managers)

Chapter 3 showed Mei thinking through PARIS on her own, in week zero. This chapter shows the same moment from another angle: David, her manager, sitting down with her before the work starts — not to hand her a plan, but to test the one she's already begun forming.

This is forward-looking, like Chapter 3, but no longer self-directed. It's the manager's version of what the coach did in Chapter 2 — except there, the coach was helping Mei make sense of something already finished. Here, David is helping her pressure-test something that hasn't happened yet, while there's still time for it to matter.

Handing over a task versus coaching through one

David could simply tell Mei: "You're leading Taiwan. Eight weeks. Let me know if you need anything." That's not wrong, exactly — but it treats Problem, Result, and Skills as settled, when they're the parts most worth checking together before Action begins.

Instead, David sits down with Mei early, and uses roughly the same structure she used on herself in Chapter 3 — but as questions, not statements.

Problem — "What do you think you're actually solving here?"

Mei gives David her week-zero answer: beat the competitor, prove Taiwan can work, protect the East Asia expansion. David doesn't correct her — she's largely right. But he adds something she doesn't have visibility into yet: leadership is watching this launch more closely than she may realize, partly because of a previous market entry that went poorly two years ago. That's not a detail Mei could have known on her own, and it changes how she'll want to communicate progress along the way — more visibly, and earlier, than she might have otherwise planned.

Result — "What does 'done well' actually look like to you? And does that match what I'd call success?"

Mei shares her draft: on time, adoption near the German numbers, no regulatory surprises. David agrees, but adds a piece Mei hadn't weighted heavily enough — leadership will care almost as much about *what Inkwell learns* from this launch, since it's meant to be a template for the rest of the region. A "successful but unrepeatable" launch — one that worked because of heroics rather than a process — would be a weaker result than Mei's draft suggests, even if the numbers looked the same.

Action — "What's the plan, and where are you least confident in it?"

This is where David does the most useful work, and it's the part Chapter 3 deliberately left thin. Mei walks him through her rough plan: get the engineering relationship going early, bring in legal sooner rather than later. David doesn't rewrite her plan — but he asks where she's least sure of herself in it. Mei admits she's never coordinated closely with an overseas engineering team before, and isn't sure how to structure that relationship so it doesn't just become a bottleneck of waiting for replies across timezones. That single admission becomes the most concrete thing David can actually help with — not by doing it for her, but by connecting her with someone on his own team who's run distributed projects before.

Skills — "What do you already have, and what should we make sure you get?"

David affirms what Mei already has — she's organized, she's shown good judgment under pressure before. But he's also direct about the gap she named herself: distributed team coordination. Rather than leaving it as a private worry, they agree on something concrete — a short call with a colleague who's done it before, in week one, not week five.

Insight — "What are you assuming that I might see differently?"

David doesn't ask Mei what she's learned — there's nothing to learn yet. Instead, forward-facing Insight becomes a check on assumptions, the same way it did in Chapter 3, except now it's a second set of eyes doing the checking. Mei repeats her assumption that "localizing" mostly means translation. David doesn't know this is wrong either — nobody does yet — but he asks a question that plants a seed: "Have you tested that assumption with anyone who's actually used the product in Traditional Chinese?" He hasn't corrected her. He's made it more likely she'll find out for herself, sooner.

What changed by having this conversation

Nothing about the underlying Problem changed by David asking these questions — Taiwan is still Taiwan, eight weeks is still eight weeks. What changed is that Mei is walking into week one with a clearer Result (one that includes "repeatable," not just "successful"), a plan that's already been pressure-tested instead of just self-approved, and a concrete next step on the one gap she was most nervous about and least likely to raise on her own.

That's the real value a manager adds here — not authority, and not better answers than Mei's own. Just a second set of questions, asked early enough to still change what happens next.

Try it

Before your next 1:1 with someone starting something new, resist the instinct to just check the plan for gaps. Ask them to walk you through their Problem and their Result first — in their own words — before you look at their Action at all. Notice what you learn about how they're thinking, not just what they're planning to do.

Chapter 5 — Defining It Together, Revisiting It as You Go (Teams)

The first four chapters all lived inside one person's head — even Chapter 4, where David and Mei talked it through together, was still fundamentally about clarifying Mei's own plan. This chapter is different in kind, not just in degree. It's the same eight weeks, but now PARIS belongs to a group: Mei, a local operations lead, a designer, and Klaus, the engineering lead dialing in from Germany.

And it introduces something the earlier chapters could only gesture at: PARIS isn't just something you define once and follow. It's something a team keeps checking itself against — and sometimes has to redefine — as the work unfolds.

Week one: defining PARIS together

David's introduction, and Mei's own week-zero thinking, gave the team a starting point — but a plan that lives in one person's head isn't the same as one a team has actually agreed on together. So in week one, before any deep work begins, Mei brings the team together to define their shared PARIS.

They don't spend long on Problem — Mei's version from Chapter 3 holds up well, and the team adds one thing she couldn't have known alone: Klaus mentions that the German version of the product had a similar-sounding input issue early on, dismissed at the time as a minor bug. Nobody had connected it to what Taiwan might need. That single comment turns out to matter more than anyone in the room realizes yet.

They agree on Result together — not just "launch on time," but Mei's fuller version from Chapter 4: on schedule, strong adoption, and something the team can point to as a repeatable process for the next market. Making this a team agreement, not just Mei's private goal, matters: everyone now has a stake in "repeatable," not just "done."

Action gets divided concretely: local operations owns the regulatory relationship, the designer owns testing the interface with real Taiwan-based users, Klaus owns technical feasibility from Germany, and Mei coordinates the whole thing and owns communication upward to David and leadership.

Week three: the Problem changes

Three weeks in, the designer's user testing turns up something nobody had named in week one: the predictive text and search aren't just slightly off — they're built entirely around Latin-alphabet input, and adapting them for Traditional Chinese isn't a tweak, it's close to a rebuild of that part of the product.

This is the moment the loop from Part 1 stops being a concept and becomes something the team has to actually do. If they treated Action as fixed once agreed in week one, this discovery would just be a delay to manage. Instead, the team goes back to Problem: the real complication was never "translate the interface," it was "the input experience assumes an alphabet this market doesn't use." That's a materially different Problem than the one they defined in week one — bigger, and more specific.

Going back to Problem changes Result, too. "Launch on time" is suddenly in real tension with "launch something that actually works." The team has an honest conversation — not a

comfortable one — about which matters more, and lands on a revised Result: launch on time with a genuinely usable, if narrower, input experience, rather than delay the whole launch to build everything they'd originally imagined. That's a real trade-off, made consciously, instead of a schedule slipping quietly without anyone deciding it should.

Action shifts accordingly. Klaus's team narrows scope to the input features that matter most, rather than everything on the original list. The designer keeps testing, now against the narrower version. Mei updates David — not to report a failure, but to explain a decision the team made deliberately, with reasoning David can evaluate rather than just accept.

Week six: checking Insight and Skills mid-project, not just at the end

Most teams save "what have we learned" for a retrospective after everything's over. This team doesn't wait. In week six, with two weeks left, Mei asks the team a version of the Insight question early, on purpose: what are we now seeing that we didn't see in week one?

The answer is the same one Mei will eventually tell in her interview a year later, but here it belongs to the whole team, in real time: they'd assumed localization meant language, and it took actual user testing to reveal it meant something closer to the product's foundations. Saying this out loud in week six — rather than only in a debrief after launch — changes how the team spends its last two weeks: they stop treating the remaining work as "fixing a bug" and start treating it as "the real work we underestimated in week one."

On Skills, the team is equally honest. Klaus admits the German team hadn't fully appreciated how different the localization challenge was until this project forced the question — a gap in the wider organization's knowledge, not just this team's. That becomes something worth carrying forward past this one launch, not just fixing quietly and moving on.

Week eight: launch, and one more loop

The launch happens on schedule, with the narrower but genuinely usable input experience the team chose in week three. It's not the launch anyone imagined in week one — it's a better-tested, more honestly-scoped version of it, arrived at because the team was willing to revisit Problem and Result mid-project instead of only defending the original plan.

Before closing out, the team does one more short PARIS loop — not a retrospective in the usual sense, but a forward-looking one: what would we tell the team taking on the next East Asia market? The answer becomes the seed of an actual playbook — starting with a Problem statement leadership hadn't originally understood, and a plan that no longer needs to rediscover, market by market, what this team learned the hard way.

Why this is the clearest case of "loop, not line"

Of all five roles in this booklet, a team project shows the loop most visibly, because a team can't quietly revise its thinking the way an individual sometimes does without noticing. Redefining Problem or Result out loud, in front of other people, requires the team to actually name what changed and why — which is exactly what makes PARIS valuable here: not as a form to complete once, but as a shared language a team can keep returning to, checking itself against, as the real shape of the work reveals itself.

Try it

If you're leading a team project, don't wait for the retrospective to ask what's changed. Partway through — not at the start, not at the end — ask the team out loud: is the Problem we named still the real one? Is the Result we agreed on still the right target? Make space for the answer to be no.

If You'd Like Help Using This

Everything in this booklet works on your own, with a notebook and a few honest minutes. But some of it — especially the coach and team chapters — is easier the first few times with someone else in the room.

If you're a **coach or manager** wanting to get better at asking the kind of questions David and Mei's coach asked — the ones that surface a Problem someone's undersold, or an Insight they haven't quite found yet — this is something I help people build, one real conversation at a time.

If you're a **team or team leader** wanting to run a genuine PARIS session — defining a shared Problem and Result at the start of a project, and knowing how to revisit it honestly partway through, the way Mei's team did in week three — I facilitate these sessions directly, so the first time your team does this, it's guided rather than improvised.

And if PARIS has been useful to you as a way of thinking — whether you're preparing for an interview, coaching someone else, or just trying to see a challenge more clearly — I'd love to hear how you've used it.

Get in touch:

www.career-factory.co

hello@career-factory.co