



LUXE BUSINESS BACKEND™ · FOUNDER FIRE CODES™

THE RAPID IMPLEMENTATION GUIDE

Your first playbook, built and live, in seven days.

BEFORE YOU READ

© 2026 TIFFANY LOPEZ · LUXE BUSINESS BACKEND™ · ALL RIGHTS RESERVED.

This guide is part of The AI Fire Codes for Founders and is licensed to the original purchaser for use inside their own business. It may not be resold, redistributed, republished, taught as your own, or rebuilt — in whole or in part — as a product, program, template, or AI tool for anyone else. The Fire Codes Method™ and the Founder Fire Codes™ framework are the intellectual property of Tiffany Lopez and Luxe Business Backend™. Examples are composites, not client stories. Nothing here is legal, financial, or HR advice.

WHAT'S INSIDE

01 How This Week Works

The ground rules, the total time bill, and what to have open.

02 Day 1 — Capture and Name

20 minutes. The incident on paper, the fire named and priced.

03 Day 2 — Trace the Dependency

20 minutes. Primary and secondary, written down.

04 Day 3 — Draft with the GPT

45 minutes. Response and prevention plans, drafted and pushed back on.

05 Day 4 — The Two Checks and Hand-Edits

30 minutes. Remove yourself, finish the judgment calls.

06 Day 5 — Assign the Owner

30 minutes. The briefing that makes it real.

07 Day 6 — Roll Out to the Team

One meeting. The announcement and the redirect.

08 Day 7 — Set the Review Cadence

15 minutes. Baseline logged, review booked, done.

09 The 30-Day View

Weeks two through four: watch, review, retire.

01 HOW THIS WEEK WORKS

Seven days. About three hours of your time, total. One fire, from burning to handled.

7

DAYS — ONE SHORT SESSION EACH

3 HRS

TOTAL FOUNDER TIME, CAPPED BY
DESIGN

1

FIRE — PICKED SMALL ON PURPOSE

This guide exists because the method guide teaches you everything, and "everything" is a terrible first assignment. Your first playbook shouldn't be your best one — it should be your fastest one, because the founder who ships a decent playbook in a week builds a second one. The founder polishing a perfect playbook in a doc somewhere builds zero.

THE WEEK AT A GLANCE — SEVEN DAYS, SEVEN OUTPUTS



Days 1–4 are desk work, alone. Days 5–7 are people work. Nothing goes public until it has passed your checks.

Three ground rules for the week:

- ◆ **Pick a real fire, not your worst fire.** Choose something that recurs monthly or more, annoys you reliably, and doesn't touch your scariest client. Your worst fire deserves the method after you've practiced the method.
- ◆ **Respect the time budgets.** Each day's session has a cap. When the timer goes, ship what you have and move on. The budgets aren't estimates — they're limits, and they're the reason this gets finished.
- ◆ **Days are moves, not dates.** Run it as a literal week if you can — momentum is a real ingredient. But if day 6 has to wait for Monday's team meeting, fine. Just don't let a "day" quietly become a month.

Have three things open all week: the Incident-to-Protocol Worksheet from your order (it mirrors this exact sequence), the Fire Codes GPT (access instructions shipped with the product — or your

own AI loaded with the master prompt from the Prompt Pack), and one doc where this playbook will live.

A word about the shape of the week, because it's built deliberately lopsided. The first four days are all you, alone, at a desk — short sessions, thinking work, nothing announced. The last three days are all people — the owner, the team, the calendar. That split is intentional: nothing goes public until it has passed your checks, so you never have to walk anything back in front of your team, and your team never watches you think out loud about a half-formed system. By the time anyone hears about the playbook, it's finished, checked, and has an owner attached. You announce conclusions, not intentions. That's a small thing that buys a lot of credibility, and credibility is the currency every playbook after this one will spend.

IF YOU ONLY READ ONE PARAGRAPH

The week works because every day ends with something physical: a capture, a trace, a draft, a checked draft, a briefed owner, an informed team, a booked review. No day ends with "thought about it." That's the whole trick.

02 DAY 1 — CAPTURE AND NAME

TIME: 20 MINUTES

OUTPUT: A WRITTEN CAPTURE + A NAMED, PRICED FIRE

1 PICK THE FIRE (3 MINUTES)

The recurring problem that pulled you off course most recently. If two candidates are tied, take the one that happened this week — warm details beat important ones today.

2 WRITE THE CAPTURE (10 MINUTES)

Five answers, plainly: What happened, with specifics — times, names, numbers. Who brought it to you. What you did, step by step. What it interrupted. How many times it's happened before, best guess. One paragraph to half a page. Specific enough that it couldn't describe someone else's business.

3 RUN THE PATTERN TEST (2 MINUTES)

Three or more occurrences? Predictable? Would your team say "oh, that again"? Two yeses and it's a pattern — continue. Otherwise pick a different fire; this method is for climate, not weather.

4 NAME IT AND PRICE IT (5 MINUTES)

Give it a short, vivid name your team will recognize on sight — "The Friday Fire Drill" beats "Client Reporting Workflow Gap." Then one line of numbers: how often it happens, what it costs per occurrence in your hours. Rough is fine; this is your baseline, and day 7 needs it.

Done when: your worksheet has a capture, a name, and a one-line cost, and you could read all three to your ops lead without adding verbal footnotes.

Watch out for: the urge to pick your biggest, scariest fire "since I'm doing this anyway." Don't. Big fires have politics, client exposure, and history — all things you don't want in a practice rep. Also watch the capture turning into an essay; if you're past half a page, you're narrating instead of recording. Facts, times, names. The polish adds nothing downstream.

03 DAY 2 — TRACE THE DEPENDENCY

TIME: 20 MINUTES

OUTPUT: PRIMARY + SECONDARY DEPENDENCY, IN WRITING

1 REREAD YESTERDAY'S CAPTURE (3 MINUTES)

Work from the page, not from memory. The capture says what happened; memory says what usually annoys you. Trace the one on the page.

2 ASK THE ONE QUESTION (5 MINUTES)

At the moment this fire needed you, what exactly were you supplying? A decision? An answer? A push? An unwritten standard? A relationship? Access only you hold? Write your answer as one sentence before looking at any list.

3 MATCH IT TO A TYPE (7 MINUTES)

Open the symptom tests — the nine sentences in the method guide, or the checkbox version on your worksheet. Check every sentence that sounds like your fire, then circle the one that names what you were supplying. That's your primary. If you're torn between two, pick the one whose fix you'd bet on and note the runner-up.

4 FIND THE SECONDARY (5 MINUTES)

Ask: if the primary were fixed tomorrow, what would still leak? That's your secondary. Write the trace as two lines — primary with a why, secondary with a why. If you want a deeper read on your type before tomorrow, your chapter in the Nine-Dependency Guide is a ten-minute detour, tonight, optional.

Done when: the trace is written in the worksheet and the cause you wrote down is structural — a missing definition, extraction, or structure — not a complaint about a person. If it names a person, trace again.

Watch out for: checking five boxes. Everything sounds a little true when you're the founder — that's how being load-bearing works. The question isn't which symptoms you've ever felt; it's which one names what you were supplying *in this specific incident*. If you genuinely checked five, your fire might actually be two fires sharing a name. Split them, keep the one from your capture, and save the other's name for playbook two.

04 DAY 3 — DRAFT WITH THE GPT

TIME: 45 MINUTES

OUTPUT: DRAFTED RESPONSE + PREVENTION PLANS

1 FEED IT EVERYTHING (10 MINUTES)

Open the Fire Codes GPT and paste in the whole trail: capture, name, numbers, trace. Then add what it can't infer — your team roster with roles, your tool stack, and any hard constraints. Answer the questions it asks you honestly, including the unflattering parts.

2 ASK FOR THE RESPONSE PLAN (10 MINUTES)

The this-week half: numbered steps your team runs the next time this fire starts, one owner per step, an escalation line that isn't you.

SAY THIS TO THE GPT

"Draft the response plan: the steps my team runs the next time this fire starts, without me in the loop. Numbered steps, one owner per step from the roster I gave you, and an escalation line that names a condition and a person who isn't me."

3 ASK FOR THE PREVENTION PLAN (10 MINUTES)

The permanent half: which structures replace what you've been supplying — ownership, boundaries, escalation rules, standards, triggers — matched to your primary and secondary types, plus the handoff layers for the eventual owner.

4 PUSH BACK, TWICE MINIMUM (15 MINUTES)

First drafts are too polite and too generic. Tell it what's wrong plainly: "we don't have that role," "step 4 still routes through me — rebuild it," "what happens when the owner is out?" Then make it tighten: response plan on one page, every step starting with a verb. Stop at 45 minutes even if it's imperfect — tomorrow's session exists precisely to catch what's wrong.

Done when: both drafts are pasted into your playbook doc — rough edges and all — and you've pushed back at least twice.

Watch out for: the perfection spiral. Somewhere around minute 35 you'll want "one more revision pass," and then another, because talking to the GPT is easier than moving to day 4. The drafts don't need to be right today — they need to be complete enough to check. Also watch for invented

details: if the draft mentions a tool you don't use or a role you don't employ, correct it now, because tomorrow's checks assume the cast list is real.

★ FIRE MARSHAL'S NOTE

Book day 4 in the morning if you can. The involvement check asks you to argue against your own comfort, and you'll negotiate harder with yourself when you're fresh. Tired founders approve their own reinstatement every time.

05 DAY 4 — THE TWO CHECKS AND HAND-EDITS

TIME: 30 MINUTES

OUTPUT: A CHECKED, FOUNDER-EDITED PLAYBOOK

1 RUN THE FOUNDER INVOLVEMENT CHECK (10 MINUTES)

Print or pull up both drafts, pen in hand. Highlight every place you appear — every step, review, approval, escalation. For each one: is this me because only I can do it, or because the draft was being polite? Anything without a genuine legal, financial, or strategic reason goes back to the GPT: "remove me from this step and rebuild it so it works without me." Whatever you keep, give it a deadline and a default so nothing waits on your inbox.

2 RUN THE ESCALATION CHECK (10 MINUTES)

Every step has exactly one owner — a real person or role you currently employ, no "the team." Every foreseeable failure — owner out, deadline slips, quality misses — has a named next move. Every escalation names a condition and a person, and the first stop isn't you. Fix the gaps now, with the GPT or by hand.

3 MAKE THE JUDGMENT CALLS YOURS (10 MINUTES)

Four things never ship straight from the machine: names and assignments, thresholds and dollar amounts, anything a client will read, and the voice guidance. Go through the drafts and replace every AI guess in those four categories with your actual call. When you're done, read the playbook top to bottom once. If you'd sign your name to it, it's ready.

Done when: both checks passed on paper, the four judgment categories are in your words, and the playbook is one clean document with a name at the top.

Watch out for: your own ego negotiating with the involvement check. "I should probably stay on this step, just for the first while" is sometimes wisdom and usually the dependency talking. The test: would you accept that reasoning from a manager you'd hired to make this process founder-free? If you keep a step, keep it out loud — written down, with the reason — so future-you can tell the difference between a decision and a habit that survived the audit.

TODAY IS THE WHOLE BALLGAME

Days 1–3 could be done badly and recovered. Day 4 can't. A playbook that skips these thirty minutes ships with you still wired into it — which means you'll be back here in a quarter, wondering why the fire relit. It relit because the draft kept you as the fuel.

06 DAY 5 — ASSIGN THE OWNER AND BRIEF THEM

TIME: 30 MINUTES

OUTPUT: AN OWNER WHO HAS SAID YES, OUT LOUD

1 CONFIRM YOUR PICK (5 MINUTES)

Usually the person closest to the fire — often whoever kept bringing it to you. Check two things: they have the authority the playbook gives them, and they have the capacity to hold it. If either is missing, fix that first; a playbook assigned to someone who can't carry it is a relight with a start date.

2 BRIEF THEM LIVE (20 MINUTES)

Face to face or on a call — never by task assignment. Walk them through the fire, the numbers, the full playbook, and the boundaries of their authority: what they decide alone, what escalates, and what success looks like at thirty days. Then tell them why you chose them, specifically. That sentence is yours to write, not the AI's.

SAY THIS IN THE BRIEFING

"This fire has cost us about [your day-1 number] every time it happens, and I've been the one catching it — which is the problem. I want you to own it, and I picked you because [the real reason]. Inside these boundaries, you decide — you don't need to check with me. If [escalation condition] happens, it goes to [escalation person]. Read it, push back on anything that won't work — you know this process from a side of it I don't — and tell me by tomorrow if you're in."

3 TAKE THEIR EDITS SERIOUSLY (5 MINUTES)

The person about to run a process sees holes the person escaping it can't. If they flag a step that won't survive contact with reality, that's the briefing working. Fold their fixes into the playbook today, and note who improved what — ownership starts with fingerprints.

Done when: the owner has the full playbook, their questions are answered, their edits are in, and they've explicitly said "I've got it" — those words or better.

Watch out for: the drive-by assignment. If the briefing happens in the last five minutes of another meeting, or lands as "hey, wrote this up, take a look when you get a chance," you've told the owner exactly how much this matters — and they'll weight it accordingly. The other trap runs the opposite direction: briefing them so thoroughly, with so much hovering, that the message becomes "I own

this but she's still holding it." Give the full package, answer the questions, then visibly let go. The letting go is part of the briefing.

07 DAY 6 — ROLL OUT TO THE TEAM

TIME: ONE TEAM MEETING (10 MINUTES OF IT)

OUTPUT: A TEAM THAT KNOWS WHO OWNS THE FIRE NOW

1 DRAFT THE ANNOUNCEMENT WITH AI, EDIT IT INTO YOUR VOICE (BEFORE THE MEETING)

Feed the GPT the finished playbook and ask for a short, plain announcement: what's changing, why now, who owns it, and what to do instead of the old way. Cut the corporate filler and make it sound like you on a good day. Keep it under a minute of speaking.

2 DELIVER IT YOURSELF, OUT LOUD (IN THE MEETING)

This is the rule that doesn't move: AI drafts the message, you deliver the change — personally, in the meeting, not posted in a channel and hoped about. Your voice is what tells the team this is real.

SAY THIS TO THE TEAM

"You all know [fire name] — it's happened [frequency] and it's been landing on me every time. That changes today. [Owner] now owns this end to end, and here's what I need from everyone: questions about [process] go to [owner], not to me. If you bring them to me out of habit, I'm going to smile and send you right back to [owner] — not because I don't care, but because this working without me is the whole point."

3 HANDLE THE CLIENT NOTE, IF THERE IS ONE (AFTER THE MEETING)

If clients will feel the change — a new point of contact, a new process — send the note framed as the upgrade it is: they now have a dedicated person. AI can draft the skeleton; every client-facing sentence gets rewritten by hand. Yours, always.

Done when: the team has heard it from your mouth, the redirect sentence has been said, and any client note has gone out in your words.

Watch out for: softening the redirect to spare feelings — yours. "You can still come to me, but maybe also loop in [owner]" is not a redirect; it's a fire exit propped open. The team will take the path they've always taken unless you close it kindly and completely. If saying it plainly feels harsh,

remember what you're actually announcing: you trust this person with something real, in front of everyone. Said right, the rollout is a promotion ceremony, not a door slamming.

08 DAY 7 — SET THE REVIEW CADENCE AND LOG THE BASELINE

TIME: 15 MINUTES

OUTPUT: A BOOKED REVIEW + A BASELINE ON RECORD

1 LOG THE BASELINE (5 MINUTES)

In the playbook doc, record the starting numbers from day 1: occurrences per month, founder-hours per occurrence, and the date the playbook went live. Thirty days from now, this paragraph is what tells you whether the fix worked — feelings won't.

2 BOOK THE REVIEWS (5 MINUTES)

Two calendar events, today: a 15-minute check-in with the owner two weeks out, and a 30-day review at the month mark. Booked now, while it's real — "we should check in sometime" is where playbooks go to die.

3 START THE RELIGHT LOG (5 MINUTES)

One section at the bottom of the playbook doc: Relights. Every time the fire reaches you after today — the old question, the old rescue — it gets a one-line entry: date, what happened, which layer of the playbook was thin. Tell the owner it exists and that entries are feedback on the document, never a grade on them.

Done when: baseline logged, two reviews on the calendar, relight log open. That's the week. Your first fire has a playbook, an owner, and a scoreboard — and you have a method you've now actually run.

★ FIRE MARSHAL'S NOTE

Keep the relight log wherever the owner already works — their project board, not your notes app. If logging a relight requires going through you, the log itself just became one more thing wired to the founder.

09 THE 30-DAY VIEW

The playbook went live in a week. It becomes true over a month.

THE WOBBLES TO EXPECT

Every first playbook wobbles in one of a few predictable places, so let's take the surprise out of them. **The week-two relight cluster:** two or three relights close together, usually traced to the same thin layer — typically the When (the owner didn't recognize a trigger you'd have caught on instinct) or the How (a step assumed knowledge that lived only in you). This is normal and fixable in a single documentation pass; a cluster is the playbook showing you exactly where the extraction was shallow. **The confidence dip:** around the second real occurrence, the owner brings you something they could have decided — not because the boundaries are unclear, but because they're testing whether the authority was real. Your answer sets the process for a year: "what would you do?" — then back their answer. **The bystander relapse:** a team member who missed the memo, or ignored it, routes around the owner straight to you. Redirect in front of nobody, kindly, and every time. And **the founder twitch:** you, reading the relight log, itching to "just tighten a few things" inside the owner's process. Route every itch through the owner. It's their document now; you're a very interested editor with exactly one standing appointment — the review.

WEEKS TWO THROUGH FOUR

Watch. Let the process hit real occurrences without stepping in. When the fire's trigger event happens — the launch, the renewal, the handoff — your job is to notice what you didn't have to do. Log every relight in one line; resist fixing anything mid-flight unless there's genuine damage on the table.

Review weekly. The two-week check-in with the owner, then a short touchpoint the week after: what ran without you, what relit, which layer was thin. Fix the package, not the person — a wobble means the document was missing something, and the fix goes into the document the same day. Expect the urge to take the process back around week two or three; that urge is the dependency asking for its job back. Sit on your hands.

Retire or revise. At the 30-day review, read the relight log against the baseline and make one of two calls. If the process has run clean through two or three real occurrences — retire the fire: mark the playbook stable, thank the owner where the team can hear it, and pick your next fire, because the method is warm. If it's still relighting — revise: the log will point at the thin layer (usually the

trigger, the standard, or an escalation that quietly dead-ends at you), fix it, and run another two weeks. Neither outcome is failure. The only failure is not looking.

YOU'RE DONE WHEN

- ☐ The fire's trigger event has happened at least twice since rollout, and the process ran without you both times — or the one wobble got fixed in the document, not by you doing the work.
- ☐ The owner has made at least one judgment call inside their boundaries without checking first.
- ☐ The relight log is empty for the last two occurrences, or every entry ends with a document fix.
- ☐ The baseline numbers moved — fewer occurrences, fewer founder-hours, or both.
- ☐ Old-habit questions are getting redirected to the owner by other people, not just by you.
- ☐ You've picked your next fire and started its capture.

IF THE WEEK FALLS APART

Sometimes the week breaks — a client emergency eats days 3 through 5, or the fire you picked turns out to be gnarlier than it looked. Two recovery rules. First, the restart point is wherever your last physical output is: if you have a capture and a trace, you restart at day 3, not day 1, no matter how much time has passed — reread what you wrote, update anything stale, and keep moving. The worksheet is your save file. Second, if the fire itself fought back — the trace kept splitting, the drafts kept sprawling — downgrade without shame. Park the gnarly fire, pick the simplest recurring annoyance you have, and run the week on that instead. The first playbook's job is teaching you the method; the method's job is handling the gnarly one, second. A downgraded week that finishes beats an ambitious one that stalls at day 4 — because the finished one changes what your team believes happens when you announce a new system.

THE COMPOUNDING PART

Playbook one takes a week because everything is new. Playbook three takes a couple of sessions. By playbook five, your team recognizes the shape and starts capturing incidents without being asked — and somewhere around there, you stop being the person who fights fires and become the person whose company writes codes.

TIFFANY LOPEZ

Business Systems Architect · Luxe Business Backend™
