



THE PLAYBOOK

The Handoff

How to move work off a founder's plate so that it actually stays off.

Written for the person doing the handing over, and the person receiving it.

Paul Prado Pacardo

Executive Assistant & Operations · me.kabuhayan.app

Contents

The work that came back

1 Why handoffs fail, and it is not the assistant

2 Sorting the work: what can actually move

3 The definition of done

4 Capturing a process without stopping the work

5 The authority ladder

6 The thirty-day sequence

7 The absence test

8 What the founder has to do

9 What the assistant has to do

10 When to hand something back

The point of all this

The work that came back

There is a particular kind of failure that almost every founder has experienced and almost none of them describe out loud, because it sounds like a complaint about a person they like.

They hire someone capable. They hand over a piece of work with genuine relief. For two weeks it goes reasonably well. Then something comes back slightly wrong, and rather than explain why, they fix it themselves, quietly, at eleven at night. A few weeks later they are doing the whole thing again, and the person they hired is doing something adjacent and less useful. Nobody had a conversation about it. The work simply migrated home.

The story that gets told afterwards is that the hire was not quite right. Sometimes that is true. **Far more often the hire was fine and the handoff was never engineered.**

A handoff is not an act of communication. It is a small piece of infrastructure. It has parts, it can be built well or badly, and when it is built badly it fails in the same three or four predictable ways every time. That is good news, because predictable failures can be designed out.

This playbook is written for both ends of that transfer. If you are a founder, it will tell you what you have to do that nobody else can. If you are the assistant, operations manager or chief of staff on the receiving end, it will tell you how to take work in a way that makes it stay taken.

It is deliberately not a productivity essay. It is a sequence, with a test at the end that tells you honestly whether it worked.

A NOTE ON THE EVIDENCE

Where I cite research in this playbook I have named it and stated its limits. Most published material on delegation rests on a single 2014 Gallup study of 143 Inc. 500 chief executives, and it is routinely stripped of that context and presented as a universal law. I use it once, precisely, and tell you what it does and does not show.

Why handoffs fail, and it is not the assistant

Watch enough of these and the same loop appears. It runs quietly, it runs fast, and by the time anyone notices, the founder has concluded something untrue about the person they hired.

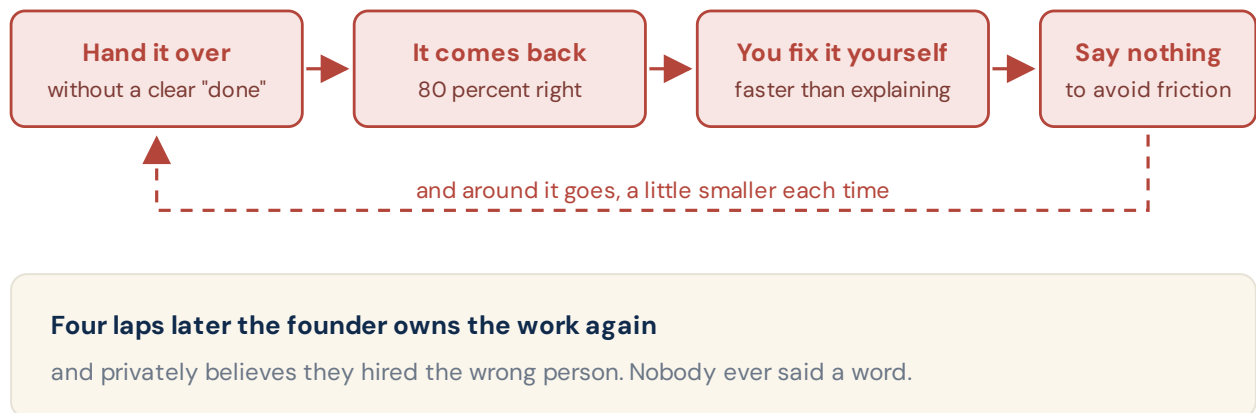


Figure: the loop that quietly returns work to the person who delegated it

Each turn of that loop is individually reasonable. Fixing it yourself really is faster this once. Saying nothing really does avoid an awkward five minutes. But the loop compounds, and it has a direction, and the direction is backwards.

Underneath it sit three causes, and none of them are about the capability of the person receiving the work.

The work was never defined, only described. "Handle the client reporting" is a description. It is not a specification. It contains no standard, no boundary and no failure condition, so the first version that comes back is judged against a picture that existed only in the founder's head.

The authority was never granted. The person was given the task but not the right to make the small decisions the task contains. So they either stop and ask, which returns the work to the founder in a different costume, or they decide and get corrected, which teaches them to stop and ask.

The correction happened silently. This is the most expensive one. A founder who fixes something without saying so has not just spent the time, they have destroyed the only feedback that would have prevented spending it again. The assistant does not know they were wrong, so they repeat it, so the founder fixes it again, and now there is quiet resentment on one side and quiet confidence on the other.

Almost every failed handoff is a definition problem wearing the costume of a people problem.

There is a reason this matters more in small companies than large ones. In a business with process infrastructure, a vague handoff hits a system that catches it. In a business of eight people, the only system is the founder, so a vague handoff simply comes back.

CHAPTER 2

Sorting the work: what can actually move

Before anything is handed over, the work has to be sorted, and the sorting is not by importance. It is by transferability, which is a different axis and produces different answers.

Most founders sort by how much they dislike a task. That reliably picks the wrong things, because the tasks people hate most are usually unusual, judgment-heavy and high-context, which makes them the hardest to transfer.

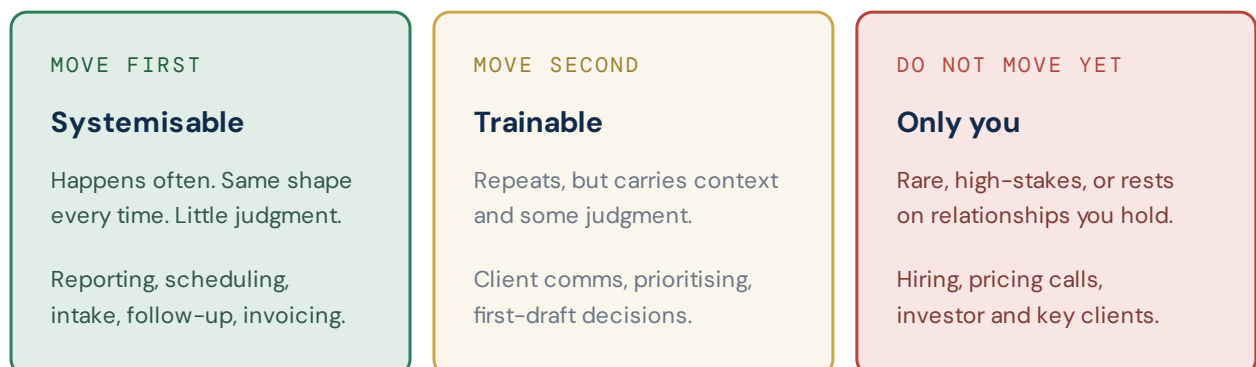


Figure: sort by transferability, not by how much you dislike it

THE HONEST TEST FOR COLUMN THREE

Ask why it cannot move. If the answer is "they would need to know X", that is a training problem and it belongs in column two. If the answer is "it has to be me", ask whether that is true or simply how it has always been. Often it is the second.

Run the sort with a real list rather than from memory. Take one ordinary week, write down everything you personally touched, and place each item in a column. The systemisable column is usually longer than expected, and at least one item you assumed was "only you" turns out to be habit rather than necessity.

Start entirely in the first column. Systemisable work transfers cleanly, produces a visible win quickly, and builds the trust that makes the second column possible. Handing over judgment work in month one is how the loop in chapter one gets started.

CHAPTER 3

The definition of done

The shortest chapter, and the one that does the most work. Almost every handoff that fails, fails here.

When you hand something over, the receiving person has to be able to answer one question without asking you: **how do I know when this is finished and correct?** If they cannot, they will produce something reasonable, you will judge it against a standard they never saw, and the loop begins.

A usable definition of done has four parts and fits in a paragraph.

01 The output

The artefact, not the activity. "A sent invoice", not "invoicing".

02 The standard

What makes it right rather than merely complete. Usually the things you have silently corrected before.

03 The boundary

What is not included, and what must be escalated rather than decided. You grant permission by naming its edge.

04 The timing

A recurring task without a stated cadence quietly becomes an occasional one.

Write it once, together, in their document rather than yours. Them writing and you correcting is worth more than you writing and them reading, because the corrections are exactly where the misunderstanding lived.

Capturing a process without stopping the work

Every founder has been told to document their processes. Almost none have done it, and the reason is not laziness. Writing a procedure from a blank page is a project, it competes with real work, and it loses that competition every single week.

So do not write from a blank page. Capture instead.

The method that survives contact with a busy business is simple. The next time the task is done, it is recorded while it happens. A screen recording with the person narrating what they are doing and, more importantly, why they are doing it that way. Ten minutes of that contains more usable knowledge than an hour of trying to remember the steps afterwards.

Then the person receiving the work writes the procedure from the recording. Not the founder. **The person who has to follow it is the person who should write it,** because they will write down the things an expert has stopped noticing, and their questions reveal exactly which steps were never actually explicit.

The founder's job is then reduced to the thing only they can do: reading a draft and correcting it. That takes ten minutes and it happens, where writing from nothing takes a morning and does not.



Correcting a draft is a ten minute job. Writing one from nothing is a project that never starts. That difference is the entire trick.

Figure: capture, then correct. Never write from a blank page.

The authority ladder

Handing someone a task without handing them the authority inside it is the most common way a handoff quietly fails. The work leaves your plate and the decisions do not, so you are interrupted about the thing you thought you had delegated, and the interruptions feel like proof that delegating does not work.

Authority is not binary and it should not be granted all at once. It moves up a ladder, one rung at a time, and both people should always know which rung they are standing on.

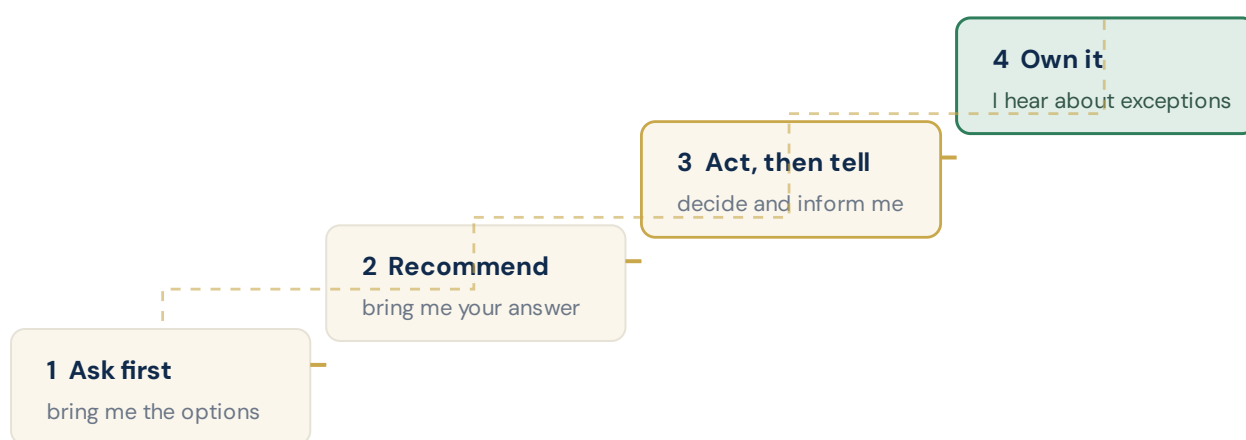


Figure: authority moves up one rung at a time, and both people know which rung it is on

The mistake in both directions is to leave the rung unnamed. A founder who has not said which level applies will be annoyed by questions at rung three and alarmed by decisions at rung one. An assistant who does not know will default to asking, because asking is safe, and safety here looks exactly like not being useful.

So name it out loud, per area, not per task. "You are at rung three on scheduling and rung one on anything a client sees." That sentence prevents a month of small frictions.

Then move the rung deliberately. A good rhythm is to review it every few weeks and promote one area at a time. Promotion should be a stated decision, not something that happens by drift, because drift produces the situation where nobody is sure who decided something.

The thirty-day sequence

Everything above assembles into a sequence. It is deliberately narrow: one workflow, thirty days, done properly. Attempting five at once is how people end up with five half-transferred processes and a founder who now has more coordination to do than before.

D 1 Sort the work and pick one

Run the three-column sort on one real week. Choose a single item from the systemisable column, ideally something weekly rather than monthly so the loop runs four times inside the thirty days.

D 2 Write the definition of done

Four parts, one paragraph, written by the receiver and corrected by the founder. Fifteen minutes, and it prevents most of what follows going wrong.

D 3 Record it being done, once

Narrated, including the reasoning. The receiver drafts the procedure from the recording afterwards.

D They do it, you watch

4-10 The receiver runs it with the founder reviewing the output before it goes anywhere. Every correction is spoken and added to the procedure. Silent corrections are forbidden in this window.

D They do it, you spot-check

11-20 Review after the fact rather than before. Move the authority rung up explicitly and say so. The procedure should stop changing around now.

D They own it

21-30 The founder hears about exceptions only. If nothing reached you in the final ten days and the work was still correct, the handoff is real.

The window that matters most is days four to ten, and specifically the rule about silent corrections. That single discipline, saying every correction out loud and writing it into the procedure, is the difference between a handoff that completes and one that quietly reverses.

A correction you did not mention is a lesson nobody learned and a cost you will pay again.

The absence test

There is only one honest way to know whether a handoff worked, and it is not a conversation. It is an absence.

Be genuinely unreachable for a week, on purpose, having told nobody it is a test. Whatever still runs correctly has been handed over. Whatever breaks, stalls or waits for you was never really transferred, and the specific point at which it broke tells you which of the four parts was missing.

HOW TO READ THE RESULT

Stalled waiting for a decision, and the authority rung was never named. Done but wrong, and the standard was never written. Not done at all, and the cadence was never stated. Done well, and you now have evidence rather than a feeling.

Most founders find the test uncomfortable for a reason that has nothing to do with the business. Being genuinely unnecessary for a week is a strange feeling for someone whose identity is bound up in being needed. That reaction is worth noticing, because it is often the real obstacle to delegation, and no amount of process design will remove it.

What the founder has to do

Four things, and nobody else can do any of them.

Say the standard out loud, once, in writing. The picture in your head is not available to anyone else. Every hour spent making it explicit is repaid several times over in corrections that never happen.

Correct out loud, every time, for the first two weeks. This is the hardest one, because fixing it silently is genuinely faster in the moment and feels kinder. It is neither, over thirty days.

Name the authority rung, and move it deliberately. Ambiguity here produces either interruptions or overreach, and you will blame the person for whichever one you get.

Then actually stop doing it. Including the version where you "just check". Checking is doing, with extra steps and worse feelings.

What the assistant has to do

The other half, and it is not passive. A handoff has a receiver, and the receiver can make it work or let it fail.

Ask for the definition of done before you start, not after you are corrected. The most useful sentence in this work is "so I know I have got it right, what does finished look like to you?" Ask it at the beginning, when it costs nothing.

Write the procedure yourself. Do not wait to be given one. Draft it from watching, send it for correction, and you will have converted a vague instruction into a checkable artefact that also happens to prove your value.

Surface the corrections you notice. If something you sent came back changed, say so and ask why. It feels like inviting criticism. It is actually the fastest way to stop needing it.

Ask for the next rung explicitly. Authority is rarely offered; it is granted when requested with evidence. "I have run this eight times without an issue, I would like to stop checking with you on it" is a reasonable and professional sentence.

Your job is not to be given work. It is to make work stay given.

When to hand something back

Not everything should transfer, and pretending otherwise damages the relationship more than admitting it early. Some work genuinely belongs with the founder, and some belongs elsewhere entirely.

Hand it back when the task turns out to be a decision in disguise, when it depends on a relationship that cannot be transferred yet, or when it happens so rarely that documenting it costs more than doing it. Say that plainly rather than struggling quietly, because struggling quietly is how the loop in chapter one restarts from the other direction.

A handoff returned deliberately, with a reason, is a professional act. A handoff returned by drift is the failure this whole playbook exists to prevent.

The point of all this

Gallup studied 143 chief executives of Inc. 500 companies in 2014 and found that those with high delegator talent generated meaningfully more revenue than those without it, eight million dollars against six. It is a real finding and it is quoted constantly, usually stripped of the fact that it looked at 143 unusually fast-growing American companies more than a decade ago. I would not build a strategy on it.

What it does suggest, and what matches everything I have watched inside real businesses, is that the constraint on a small company is very often the number of things that must pass through one person. Delegation is the only lever that moves that constraint, and delegation fails far more often for structural reasons than for human ones.

Which is oddly good news. A person you cannot change is a hard problem. A handoff you did not engineer is a Tuesday afternoon.

Pick one weekly task. Write four sentences describing what done looks like. Record it once. Correct out loud for two weeks. Name the rung. Then leave for a week and find out the truth.

That is the whole method, and it works whichever end of it you are standing on.

Paul Prado Pacardo

Senior Executive Assistant, Operations and Project Manager.
Ten years supporting C-level leadership. Available for remote work.

me.kabuhayan.app
linkedin.com/in/paulpacardo
pacardopaul18@gmail.com