TOOLKIT — FIELD GUIDE

The Stack Assessment.

Five layers, in an order that is not negotiable, and the uncomfortable question of how much of your effort is sitting on top of a gap.

Companion to the Stack Assessment workbook.
Start here. It tells you which of the others you need.

CNNCTD, LLC — Working. Better.

cnnctd.work

This is a sequence, not a score.

Nobody is grading you and there is no prize for being on layer five. The only output that matters is which single layer to work on next, and the reason the others are listed at all is to make that one obvious.

Most operating assessments produce a radar chart with five spokes and an implied instruction to improve all of them. That is not how operating capability works. The layers sit on each other. Foresight with no cadence has nowhere to be reviewed, so the lists rot. A cadence with no visible work spends its time reconstructing what happened. Distributed ownership without any of it is delegation into a vacuum.

> **Working on three layers at once reliably produces three layers that do not hold. This is the single most common way good operating intentions produce nothing.**

The number to read twice

Layers you are maintaining on top of the gap. The workbook finds the lowest layer that is not holding, then counts how many layers *above* it are nonetheless being maintained.

Every one of those is real effort, genuinely spent by capable people, producing less than it should — and it will keep doing that however well it is run, because the problem is underneath it. Most businesses have at least one. It is very often the thing they are proudest of.

> **Answer for this month.**
>
> Not for how things are supposed to work, and not for the good quarter. Yes means true almost always. Partly means true when nothing else is on fire. No means what it says. The file is only as useful as the least flattering answer you are willing to give it.

Each one is the substrate for the next.

The order is not a maturity model borrowed from somewhere. It is a dependency chain, and each link fails in a specific way when the one below it is missing.

LAYER	WHAT IT MEANS	HOW THE ONE ABOVE FAILS WITHOUT IT
1 · Personal discipline	The operator's own commitments hold without being chased.	A cadence run by somebody who cannot hold their own weekly commitment will not hold a team's. It survives about five weeks on enthusiasm.
2 · Visible work	All active work is in one place, and it is current.	The cadence spends its time reconstructing what happened out loud, from memory, badly — which is how a fifteen-minute touch becomes fifty.
3 · Operating cadence	A rhythm exists whose output is an updated record.	Foresight has nowhere to be reviewed. Risks and assumptions get written once and never read, which is why imposed RAID logs always rot.
4 · Foresight	What might go wrong is written down before it does.	Delegation without it means handing someone a decision with no view of what is coming — so they escalate, correctly, and the delegation reverses.
5 · Distributed ownership	The business does not queue at one person.	Nothing above this. It is the outcome the other four exist to make possible.

Layer one is a person, and that is deliberate

It is the only layer about an individual rather than a system, and it is there because the ceiling idea is fractal. A team's cadence cannot exceed the reliability of the person running it, in exactly the way a stacked habit cannot exceed the reliability of its anchor.

This is also the layer people skip when scoring, because it feels like a judgement about somebody. It is not. It is a question about whether the right person is holding the thing, and getting that wrong quietly is far more expensive than asking it out loud once.

The most expensive work in a business is competent work on the wrong layer.

Not wasted work — competent work, done by good people, on something that cannot pay off because the thing underneath it is not holding.

The pattern is easy to recognise once named. A business runs a genuinely decent weekly meeting. It has started pushing decisions down. Both are real efforts and both are being run properly. And nobody can reliably say what is in flight this week, because work lives in three places and none of them is current.

Everything above that gap underperforms. The meeting is good and produces less than it should, because half of it goes on establishing what is true. The delegation is sincere and keeps reversing, because the person you delegated to cannot see enough to decide. Neither failure looks like a visibility problem from the inside. They look like the meeting needing a better agenda and the person not being ready.

> **A layer built on sand does not fail loudly. It underperforms quietly, for years, while everyone involved works hard and concludes the problem is somewhere else.**

Why the instinct is to build higher

Because the lower layers are boring and the higher ones are interesting. Nobody wants a project called "get all the work into one place". People want to talk about decision rights and risk posture. So the visible layer gets skipped, and the interesting work goes on top of the gap.

It is also because the lower layers feel done. "We have a project tool" reads as visible work solved. The five statements on that layer are written to separate having a tool from having visibility — is it current *today*, does every item have one named owner, would somebody outside the team understand it, do people check it instead of asking. Most businesses with a tool score four out of ten.

What to do about the layers above the gap

Not stop them. Stopping a functioning cadence to go and fix visibility loses you the rhythm and buys nothing. Keep them ticking over at their current level and **stop investing in them** — no new ceremony, no improvement projects, no expanding scope — until the layer underneath holds. Then they will improve on their own, without you doing anything to them, which is the clearest evidence available that the diagnosis was right.

Fifteen minutes, and one trick worth using.

Twenty-five statements, three possible answers. The mechanics take a quarter of an hour. Getting a useful answer out of it takes one additional decision.

Have two people fill it in separately

THE TRICK

The founder and whoever would actually run the thing. Do not discuss it first. Then compare.

Any layer where you differ by more than two points is more informative than either score, because it means one of you is looking at the intent and the other at the practice. That gap is the finding. It is also, in our experience, the most productive forty minutes available to a leadership team that thinks it agrees about how the business runs.

Notice which statement you argue with

THE TELL

There is usually one where you find yourself constructing a case for Partly. That construction is the answer — nobody builds an argument for a statement that is comfortably true.

The Snapshot sheet asks which one it was, and it is the question we care most about. It is almost always one layer lower than people would like it to be.

Things that skew the result

- **Answering for the best team rather than the business.** One function running beautifully does not make the layer hold. Score the weakest thing that matters.
- **Answering for the tool rather than the practice.** Owning software is not the same as the statement being true. Every statement is deliberately about behaviour.
- **Answering after a good week.** If you have just come off a strong sprint, wait. The file is measuring the ordinary month, not the best one.
- **Lowering the threshold.** The Thresholds sheet lets you move what counts as holding. Moving it down until your sequence looks better is the only genuinely useless thing you can do with this file.

Redo it quarterly.

Fifteen minutes, four times a year, and the interesting output is not the scores — it is whether the layer marked START HERE has changed. If it has not moved in two quarters, the thing you are doing about it is not working, and that is worth knowing well before the year is out.

What to do with a sequence once you have one.

The workbook names one layer and one file. The temptation is to treat that as a starting point for a broader programme. Resist it — the narrowness is the whole value.

WEEK 1

Read the sequence out loud to the people affected.

Including the part about layers above the gap, and especially if one of them is something the team is proud of. Saying "this is good work and it is underperforming because of something below it" is a very different message from "this is not working", and the first one keeps people with you.

WEEKS 2—8

One layer. Use the file it names.

Six to ten weeks is realistic for a layer to hold under pressure — not to exist, to hold. Anything faster than that is usually the layer existing rather than working, which is precisely the distinction the scoring is built to catch.

WEEK 9

Reassess. Fifteen minutes.

Two outcomes are useful. The layer now holds and the sequence moves up — good, and now the layers above should improve without being touched. Or the score has not moved, which means the approach was wrong rather than the diagnosis, and that is worth catching at week nine rather than month nine.

ONGOING

Expect to go backwards sometimes.

Layers un-hold. People leave, a quarter gets brutal, the person running the cadence gets promoted. The sequence changing is the file working, not the effort having been wasted.

> **If layer one is the gap, that is a different conversation.**
>
> Personal discipline is the only layer that is about a person rather than a system, and the answer is usually not "try harder" — it is that the wrong person is holding it, or the right person has no authority to change how the team works. Both are fixable. Neither is fixed by a workbook.

The front door, not a fifth tool.

Four of the five layers have a working file behind them, and this assessment exists to stop you guessing which one to open. Personal discipline has the Stack Builder. Visible work and operating cadence both run off the Operating Cadence workbook — the Board sheet first, then the rest. Foresight has the RAID Log. Distributed ownership has the Founder Dependency Audit.

Used in the wrong order they still work, individually, and they will still under-deliver — which is the same failure the assessment is describing, applied to our own toolkit. A RAID log handed to a team with no cadence rots on schedule. An audit of founder dependency in a business where nobody can see the work produces a list nobody can act on.

All of it is **Discipline Stacking**: capability built by layering a small number of practices that hold each other up, rather than adopting a framework wholesale and hoping the culture arrives with it. The literal stacking is the point — each layer that holds makes the next one cheaper.

What we actually think

Most businesses do not have an operating problem so much as an ordering problem. The individual practices are well understood and freely available; what is missing is any honest answer to which one is load-bearing right now. That answer is worth more than any of the practices, and it takes fifteen minutes.

Send it back and we will read it before we talk.

This is the shortest file in the toolkit and the most useful one to return, because it says where you are rather than how one part is going. The Snapshot pulls your sequence, the gap, and all five scores, and asks three questions.

Email it to connect@cnnctd.work. The question worth answering properly is which statement you argued with yourself about — that one is almost always the finding.

And if your sequence says layer one is holding and everything above it is fine, we will tell you that and go away. That happens more than you would expect.

CNNCTD, LLC · cnnctd.work · We do not share this file, we do not sell it on, and we do not add you to anything.