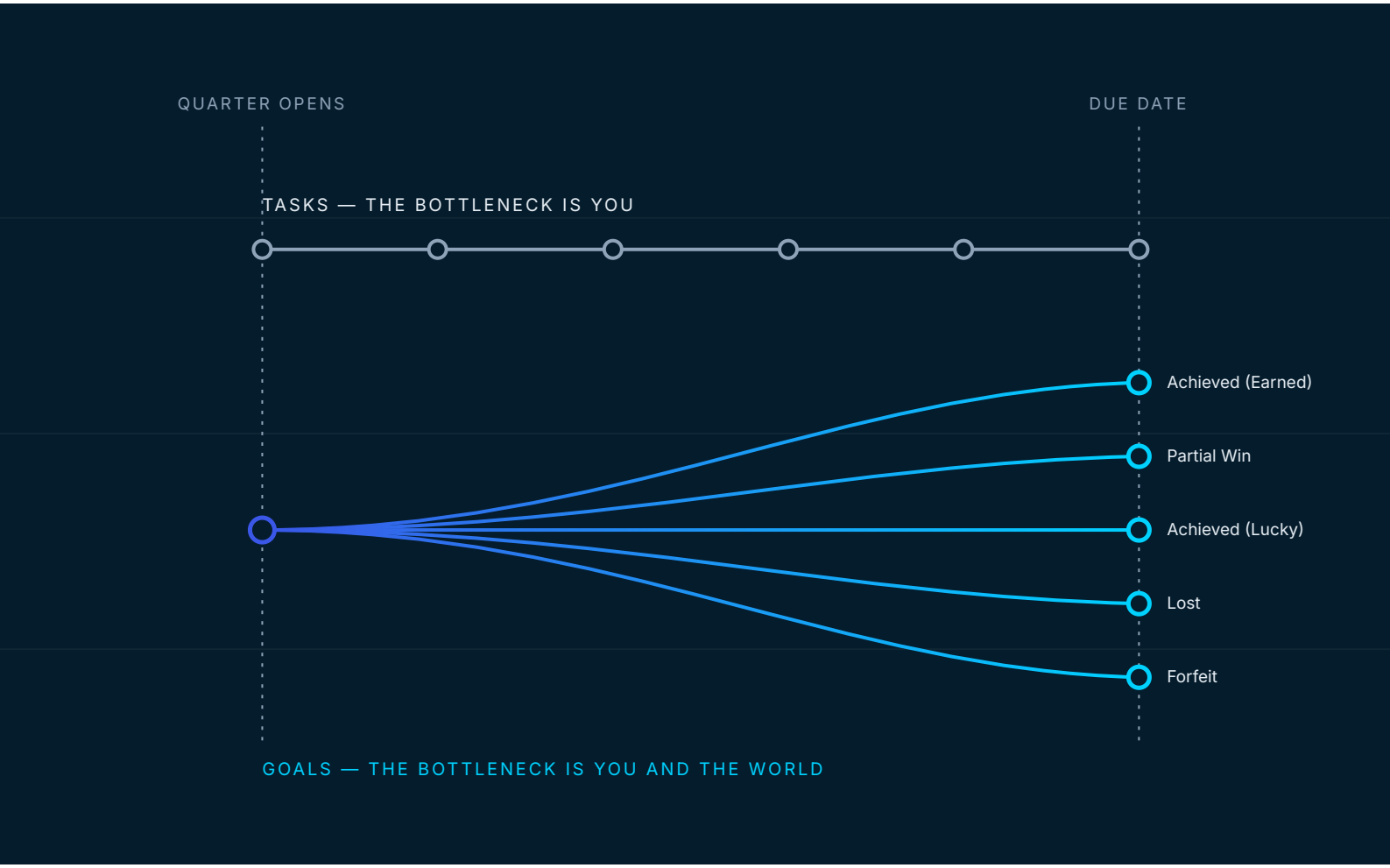
Aviral Prakash

OPERATIONS · WORK MANAGEMENT

Tasks Are Not Goals

How to build a work manager that separates what you control from what the world decides, and tells a bad bet from work that was never done.

October 2026



IN SHORT

When every piece of work is a to-do, no one can tell a bad bet from bad execution

Work comes in two kinds. A **task** is fully in your control: ship the integration, build the pricing sheet, prepare the data room. If it isn't done, it is because you didn't do it. A **goal** depends on the world as well: a hospital signing a pilot, an investor closing a round, a clinician choosing to log in. You can do everything right and still miss it.

Most tools put both in one list, with one set of statuses. When a goal is tracked like a task, a missed quarter produces one of two wrong conclusions. The team blames itself for something the world decided, or it blames the world for work it never did. Either way, the next quarter starts with the wrong move.

This report sets out, in full, the system A|P runs on every week and sets up for its clients: the idea behind it, the two trackers and their vocabularies, the arithmetic, the links between them and the review rhythm.

2	70%	90 days	5–25%
trackers with separate vocabularies: Status for tasks, Confidence for goals.	of target is the line between a Partial Win and a Lost goal.	is the longest any goal or task may live. Nothing rolls into the next quarter.	is the weight band for any one goal; an owner's goals sum to 100%.

Takeaways from this report

- **Separate work by control, not by importance.** A task has one bottleneck: you. A goal has two: you and the world. Neither is more important, but they need different tracking.
- **Track goals by confidence, not completion.** A goal is never "50% done". While it is live, record which way it is trending. At the due date, record a verdict that separates skill from luck.
- **Goals are moved by procedures, not tasks.** Tasks are for one-offs and experiments. When a task recurs, it becomes a procedure, and the system flags any goal with nothing behind it.
- **The build is the easy part.** The full data model fits on one page (Exhibit 12). The weekly review, and an honest verdict at quarter end, are what make it pay.

> HOW TO READ THIS REPORT
>
> Sections 01–03 cover goals: the idea, the tracker and its states. Section 04 covers tasks, 05 procedures and how the three connect, 06 the weekly rhythm, and 07 how to build or configure it, with suggested templates. Exhibit 14 is an audit you can run against the tool you use today.

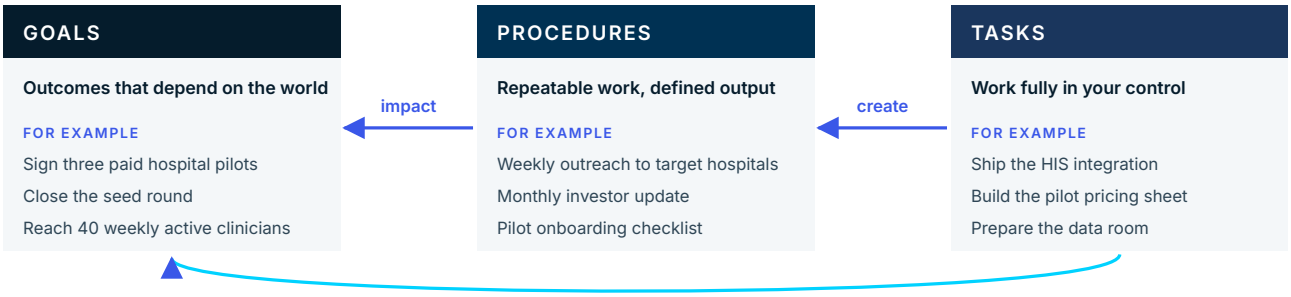
THE SYSTEM ON ONE PAGE

Three objects, two vocabularies and one weekly review make up the whole system

EXHIBIT 1

Goals, procedures and tasks, and how they connect

The three record types, with examples, and the links between them



Tasks can also support goals directly (one-offs, experiments), and a goal can come before a task

AROUND THE THREE OBJECTS · TWO RULES THAT HOLD THE SYSTEM TOGETHER

ONE QUARTER · 90 DAYS AT MOST	Every goal and task opens and closes inside it. Nothing rolls over.
WEEKLY GOAL REVIEW	Reads Confidence, Linearity and support flags; sets next week's focus.

Examples are illustrative.

OBJECT	THE QUESTION IT ANSWERS	TRACKED BY	HOW IT ENDS
Goal	Did the world give us the outcome we bet on, and did we earn it?	Confidence, plus a baseline, a target and the result so far	A verdict at the due date (Exhibit 3)
Procedure	Is the repeatable work that moves the goal actually running?	Status; how often it runs	Retired or superseded, not closed each quarter
Task	Did we do what was fully in our control, by the date we promised?	Status; Deadline and Work Date	Closed Fully, Closed w/Qualifiers or Abandoned

01

CONTROL

THE CORE IDEA

Work is sorted by how much control you have over it, not by its size, horizon or importance. A three-month build can be a task. A one-week signature can be a goal.

NO VALUE JUDGEMENT

Goals are not automatically more important than tasks. Over time, though, almost all work that matters is a goal or links to one. A standalone task should be rare.

GOALS AND TASKS

Work divides by control: you alone, or you and the world

A **task** is work where the only bottleneck is you, however you define you: a person, a team, a department, the company. It may take ten minutes or three months. You may not yet know how to do it, but you know you can learn, which means you know what to do. If a task isn't done, it isn't bad luck. You ran out of capacity, chose to do something else, or avoided it.

A **goal** is work whose completion genuinely depends on the world to a non-trivial extent: a customer, a group of customers, another company, a regulator. Working hard influences it but cannot guarantee it. You can do everything right and still miss. You can do nothing and still hit it.

EXHIBIT 2

Most goals have a task that looks like them; the difference is who has to say yes

Illustrative pairs from a healthtech company

TASK · BOTTLENECK: YOU	GOAL · BOTTLENECK: YOU + THE WORLD
Ship the HIS integration for the pilot site	Get the integration signed off by the hospital's IT team The hospital decides
Build the pilot pricing sheet	Sign three paid hospital pilots this quarter Three buyers decide
Prepare the data room	Close the seed round Investors decide
Launch the clinician app	Reach 40 weekly active clinicians Clinicians decide
Post the role and run the interviews	Hire a solutions engineer A candidate has to accept

Examples are illustrative.

The test is one question: **if the team does everything it planned, is the result guaranteed?** If yes, it is a task. If no, it is a goal. The hiring row shows the trap: running interviews is a task, but the hire happens only when a candidate says yes.

None of this says tasks and goals are unrelated. They are correlated. Tasks are just not 100% causative of goals, and that gap is what the rest of the system is built to measure.

At the due date, every goal gets one of five verdicts, and each one tells you what to do next

Two facts decide how a goal closes: whether the work behind it was done, and how much of the goal arrived. Together they produce four result verdicts, plus a fifth for goals the team chose to stop pursuing.

EXHIBIT 3

Five verdicts, decided by two facts: was the work done, and how much of the goal arrived

Goal verdict at the due date, by execution and share of the target achieved

	BELOW 70%	70% TO BELOW 100%	100% OR MORE
Work done	**Lost** The world said no, or the bet was wrong.	**Partial Win** Effectively positive; the world resisted a little.	**Achieved (Earned)** The bet and the execution both held.
Work not done	**Lost** Below the line, whatever the reason.	**Achieved (Lucky)** The outcome arrived without the work.	**Achieved (Lucky)** Even a full result is luck if nothing was run.
Dropped	**Forfeit** A decision, not a result: the team chose to drop the goal and stop working towards it.		

VERDICT	WHAT TO DO WITH IT
Achieved (Earned)	Repeat it. If tasks rather than a procedure carried it, write the procedure now (Section 05).
Partial Win	Keep the bet. Look at where the world resisted, using the evidence in a confidence log.
Achieved (Lucky)	Don't celebrate it, and don't plan on it. Treat it as a repeatable success and it will fail next time.
Lost	Read it alongside the goal's support. If the work ran, the hypothesis was wrong: change the bet. If it didn't, the bet is untested.
Forfeit	Record the decision and the reason, then decide whether anything replaces it next quarter.

SAME VERDICT, DIFFERENT NEXT MOVE · ILLUSTRATIVE

Two founders each bet on three paid hospital pilots this quarter, and each signed one: 33% of target. Both goals close as **Lost**. The first ran a weekly outreach procedure to sixty hospitals. Her bet, that mid-size hospitals buy after a demo, was tested and failed, so she changes the bet. The second never ran outreach. His goal shows as task-dependent, with most linked tasks still open. His bet was never tested, so he runs it properly before replacing it. The verdict is the same; the support data next to it tells them apart.

02

THE GOALS TRACKER

ONE GOAL, ONE BET

One goal is one bet on the world, inside one quarter. If the result is fully in your control, it's a task.

YES-OR-NO GOALS

When the outcome either happens or doesn't, such as signing one anchor customer, the scale simply runs from 0 (it didn't happen) to 1 (it did).

ONE OWNER

Every goal and every task has exactly one owner. Helpers can be many.

MEASURING A GOAL

Every goal gets a baseline, a target and a weight before it gets a status

Every goal carries three numbers:

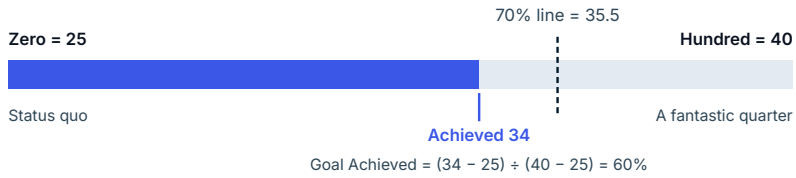
- **Zero:** what you would get under the status quo, doing nothing.
- **Hundred:** what a fantastic result looks like.
- **Achieved:** the actual result so far, updated by the owner.

Because Zero is rarely nothing, progress is measured from the baseline, not from zero.

EXHIBIT 4

Zero and Hundred put any outcome on one 0–100% scale

Illustrative goal: weekly active clinicians. Zero 25, Hundred 40, Achieved 34



Bar drawn to scale between Zero and Hundred. Example is illustrative.

Weight

Weight says how much each goal matters to its owner this quarter. No single goal may carry more than 25% or less than 5%, and each owner's goals sum to 100%. So every owner holds at least four goals and at most twenty.

A suggested template for writing up each goal, the goal page, is on page 15.

Two calculations show where a goal stands today and where it will land at the due date

Goal Achieved

HOW MUCH HAS ARRIVED

The share of the way from Zero to Hundred covered so far. It reads 0 when nothing is recorded or the result is below Zero. It is not capped at 100%, because Earned starts at 100%.

> (Achieved – Zero) ÷ (Hundred – Zero)

Linearity

WHERE YOU WILL LAND

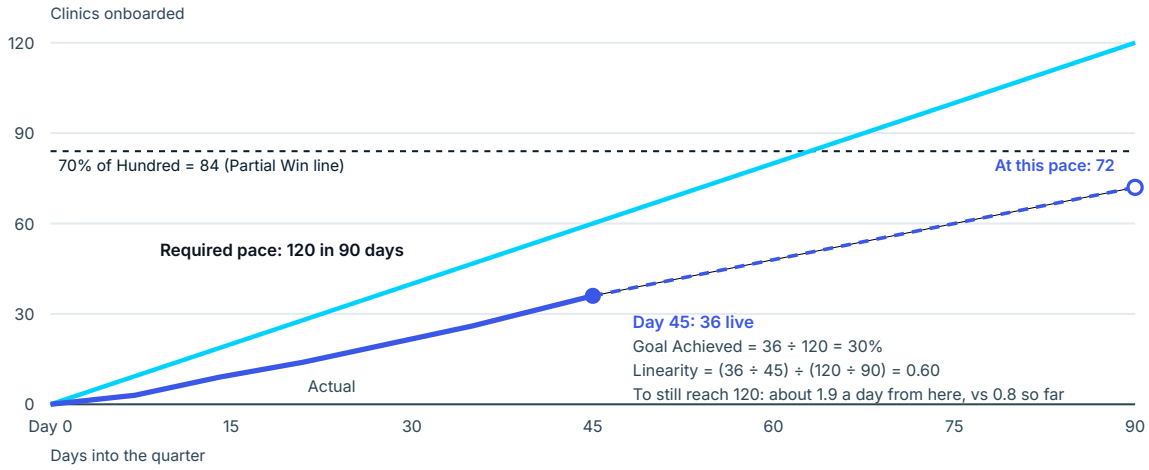
Linearity looks ahead. It compares the pace so far with the pace needed, and so tells you where the goal will end up at the due date if nothing changes. 1.0 means you will reach Hundred; 0.6 means you will finish 60% of the way there. It never goes below 0.

> [(Achieved – Zero) ÷ days elapsed] ÷ [(Hundred – Zero) ÷ 90]

EXHIBIT 5

Halfway through the quarter, 36 of 120 clinics are live, and the current pace lands at 72, below the 70% line

Illustrative goal: new paying clinics onboarded. Zero 0, Hundred 120, 90-day quarter, read on day 45



Drawn to scale. Example is illustrative.

Linearity is arithmetic; Confidence (Section 03) is the owner's judgement about the world. They can disagree: a goal at 0.6 may still be Trending to Achieved (Earned) if a large hospital group is about to sign. The weekly confidence log is where that evidence is written down.

Two rules any version needs, or Linearity misleads:

- **Levels and averages don't accumulate.** For a goal measured as a level or an average, such as weekly active clinicians or average response time, Linearity should simply equal Goal Achieved.
- **Count today as an elapsed day.** Days elapsed is 90 minus the days until the due date, plus one, so the first day of the quarter never divides by zero.

03

CONFIDENCE

WAITING IS NOT A STATE

If a client, a regulator or the weather is holding a goal up, no amount of working harder fixes it: you wait or you pivot.

CLOSING STATES

A verdict is chosen only when the due date has arrived or passed. Until then, the goal is trending.

SUCCESS METRIC

Strategic clarity: knowing which bets paid off because of the work, and which failed despite it.

LIVE GOALS

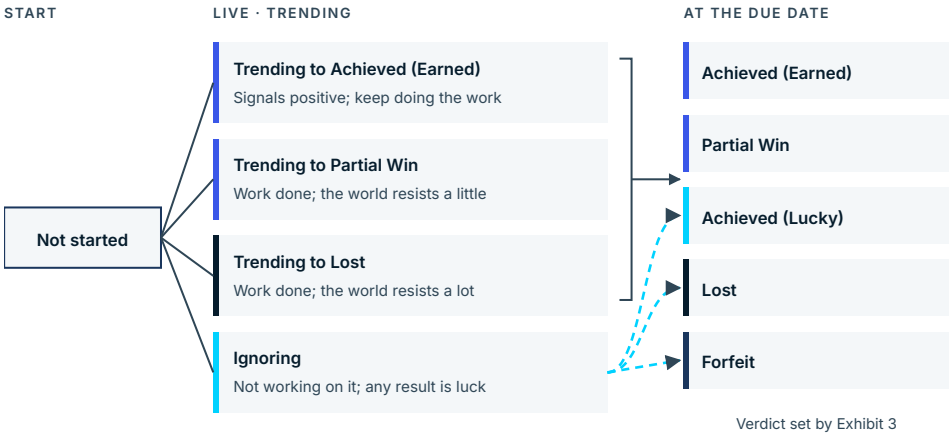
Track where a live goal is heading, not how much of it is done

A goals tracker is not a to-do list. It is a dashboard of bets. The live question is not "how far along are we?" but "given the current state of the world, how likely is this to happen?" Four live states answer it, and each one says something about both the team and the world.

EXHIBIT 6

Four live states lead to five verdicts; a goal marked Ignoring can only close as luck, a loss or a drop

Goal Confidence values, from start to close. Dashed lines: the only verdicts open to a goal whose work was not done



- **Each state is set by evidence, not mood:** reply rates, proposals accepted, pilots in procurement, a regulator's timeline.
- **Trending to Partial Win and Trending to Lost both assume the work is being done.**
- **If it isn't, the honest state is Ignoring,** and the owner should know that any result will be luck.

THE CONFIDENCE LOG SUGGESTED FORMAT

One line per weekly review: date — Confidence — evidence — next move

Illustrative: "Week 7 — Trending to Partial Win — two hospitals in procurement, the third has gone quiet — ask the medical director for a decision date."

A|P uses this format with its clients. Adapt it to your team.

04

THE TASKS TRACKER

NO RISK, ONLY DRIFT

You cannot lose a task. You can only neglect it or abandon it. The danger is drift: it gets pushed to tomorrow until it stops mattering.

TWO DATES

Deadline is the last day, and it never moves. **Work Date** is the next day you will work on it, and it moves as often as you like before the Deadline.

SUCCESS METRIC

Execution reliability: when you say you will do it, it gets done, or you consciously kill it.

EXECUTION

A task cannot be lost, only neglected or abandoned, so its statuses are about honesty, not risk

Because the only bottleneck is you, a tasks tracker needs no risk rating. Nothing in it is "at risk". What it needs is a vocabulary that makes the owner say why something isn't moving: capacity, a deliberate choice, or avoidance.

EXHIBIT 7

Nine task statuses in three groups; two of them exist to separate a choice from neglect

Task Status values, grouped as the tracker groups them

TO-DO	IN PROGRESS	COMPLETE
Not started No work of any kind yet	**WIP** In progress	**Closed Fully** Done to the extent originally planned
Blocked (Start) Another task must finish before this can start	**Blocked (End)** Another task must finish before this can finish	**Closed w/Qualifiers** Partly done and judged good enough
Deprioritised Consciously set aside for now		**Abandoned** Wilfully dropped
Being Ignored Not being done, against your own wishes: a warning sign		

- **Deprioritised** means you are managing limited time.
- **Being Ignored** means you are losing control of your schedule.
- **Closed w/Qualifiers** exists to stop perfectionism: in business, good enough is often the right call.
- **Both Blocked states** mean waiting on *another task*. If a task is waiting on someone outside the team, check whether it is really a goal.

EVERY TASK CARRIES TWO NUMBERS

Target is how many sub-tasks make it complete, or 1 for a yes-or-no task. **Achievement** is how many are ticked. Splitting sub-tasks into Must Do and Should Do sets the line between Closed Fully (everything done) and Closed w/Qualifiers (every Must Do done).

A suggested template for writing up each task, the task page, is on page 15.

05

PROCEDURES

THE CHAIN

Tasks create procedures; procedures impact goals.

WHEN TO WRITE A PROCEDURE

When a task recurs, or when finishing it shows you how that kind of work should always be done.

THE THIRD OBJECT

Goals are moved by procedures, and tasks are how you find out which procedures to write

Procedure

- A repeatable process that reliably produces a defined output.
- Run consistently, it compounds: each run a little better or faster than the last.
- The main engine for achieving goals.

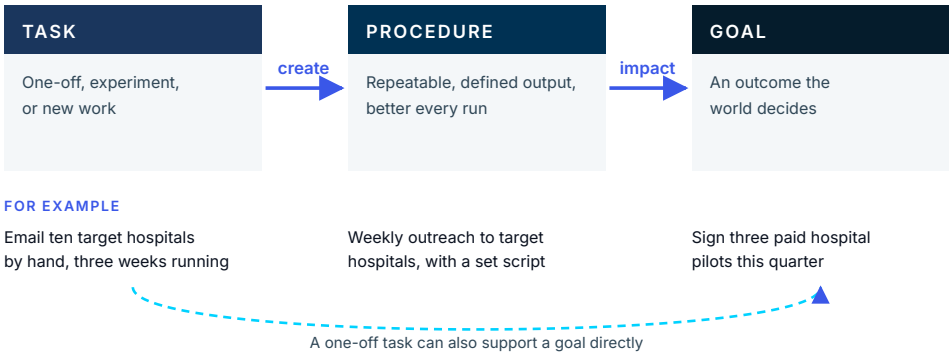
Task

- Work that hasn't been systematised yet.
- A genuine one-off, an experiment, or something too new to have a standard approach.
- Necessary, but a task that recurs should become a procedure.

EXHIBIT 8

Tasks create procedures, and procedures impact goals

The chain, with an illustrative example under each link



Examples are illustrative.

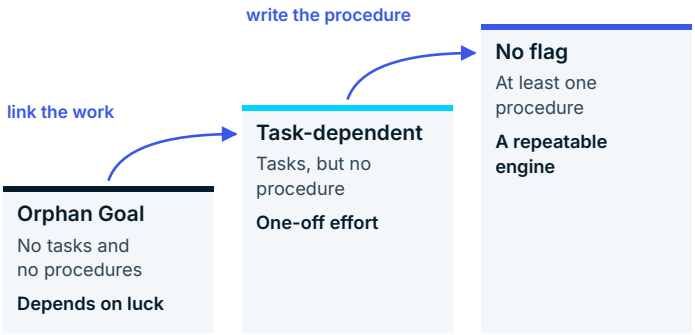
When you find yourself hitting a goal through repeated manual tasks rather than a well-run procedure, you have discovered what the procedure should look like. Write it, and link it to the goal it serves.

A goal backed only by tasks is fragile, and a goal backed by nothing is a bet on luck

EXHIBIT 9

One flag sorts every goal into three levels of support

Support flag, from the number of linked tasks and procedures



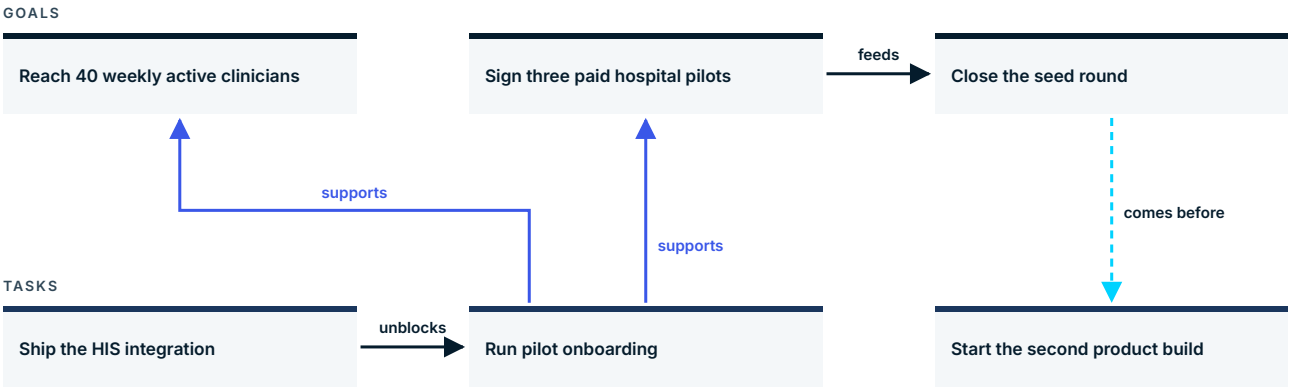
The flag turns the chain into something the weekly review can see. A goal with no flag still needs its procedure to be running. A goal with a flag has no engine to judge.

It is also how to read a Lost verdict. A Lost goal that was task-dependent, with most linked tasks still open, is an untested bet, not a disproven one.

EXHIBIT 10

Links run in every direction: task to task, goal to goal, one task to several goals, and a goal before a task

Illustrative links in one healthtech company's quarter



Examples are illustrative.

- **Task to task:** onboarding is Blocked (Start) until the integration ships.
- **One task, two goals:** pilot onboarding serves both the pilots goal and the active-clinicians goal.
- **Goal to goal:** signed pilots make the round easier, so the pilots goal sits upstream of the funding goal.
- **Goal before task:** the second product build starts only once the round closes.

> **THE QUARTER RULE**
>
> Every goal and task lives inside one quarter, 90 days at most. Longer work gets a fresh goal or task each quarter. Unfinished work is never moved forward: close the goal with its verdict or the task as abandoned, and open a new one. If a goal can be moved to next quarter, nobody ever has to admit it was missed.

06

RHYTHM

PERSONAL VIEWS

In this set-up, every view filters to "owner is me". The data is shared; each person's screen shows only their own commitments.

THE WALL

After the review, the week's focus and the nearest deadlines go on one shared page the whole team can see.

THE REVIEW IS A TASK

The weekly review is itself a recurring tactical task.

VIEWS AND REVIEW SUGGESTED SET-UP

Four saved views and one weekly review turn tasks and goals into a working rhythm

This is the set-up A|P uses with its clients; your team can design its own. Each owner works from one dashboard of four views, each answering a single question, and the weekly review reads the goals view line by line.

EXHIBIT 11

Each view answers one question, and two of them read the same tasks by different dates

The owner's dashboard: views, filters and what each is for

VIEW	BUILT FROM	QUESTION IT ANSWERS
Deadlines missed	Tasks: owner is me; status in To-do or In progress; Deadline on or before yesterday	What did I promise and not deliver?
Deadlines this month	Tasks: owner is me; calendar by Deadline	What can't move?
Weekly work plan	Tasks: owner is me; one-week calendar by Work Date; sorted by Status	What am I doing this week?
Goals this quarter	Goals: owner is me; current quarter; grouped by Confidence; sorted by due date, then weight. Shows Linearity, Goal Achieved, Zero, Hundred, Achieved, Weight, procedure count, support flag	Which bets are working?

THE WEEKLY GOAL REVIEW SUGGESTED BY A|P

1. Open every goal in the quarter.
2. Sort them into four boxes: going well, improving, not going well, slipping.
3. For each goal, set Confidence from the evidence and add one line to its confidence log.
4. Check the support flags. Orphan and task-dependent goals need an engine, or an honest Ignoring.
5. Set the focus for the coming week on the shared wall, and move Work Dates to match. Deadlines stay where they are.

The steps A|P runs with clients. Adapt the order and the boxes to your team.

Two calendars over the same tasks is deliberate. The Deadline calendar is the record of promises; the Work Date calendar is the plan, and the plan is allowed to change.

The whole data model on one page: three record types, their fields and the rules they enforce

EXHIBIT 12

The specification: every field, its type and the rule it enforces

FIELD	TYPE	RULE
GOAL		
Name, ID	Text; auto ID	One bet, phrased as an outcome
Owner; Helper(s)	1 person; people	Exactly one owner; any number of helpers
Due date	Date	Inside the current quarter
Zero; Hundred; Achieved	Numbers	Status quo; a fantastic result; actual to date. Yes-or-no goals: 0 and 1. Hundred ≠ Zero
Metric type	Select	Cumulative, or level/average (sets how Linearity is calculated)
Weight	Percent	5–25% per goal; an owner's goals sum to 100% each quarter
Confidence	Status, 3 groups	Not started · four live states · five verdicts (Exhibits 3 and 6); verdicts only at or after the due date
Related tasks, procedures, goals	Two-way links	A task or procedure can serve several goals
Goal Achieved	Calculated	(Achieved − Zero) ÷ (Hundred − Zero); 0 if blank or below Zero
Linearity	Calculated	Pace so far ÷ pace needed, floored at 0; level-type goals use Goal Achieved
Linked tasks progress	Roll-up	Share of linked tasks in the Complete group (count Abandoned separately)
Support flag	Calculated	"Orphan Goal" if no tasks and no procedures; "Task-dependent" if tasks only; blank otherwise
Quarter, week, days left	Calculated	From the due date; drive the views
TASK		
Name, ID, owner, helpers	Text; ID; people	Exactly one owner
Type	Strategic / Tactical	Tactical tasks link up to strategic ones
Status	Status, 3 groups	Nine values (Exhibit 7)
Deadline	Date	Inside the quarter; never moved once set; reminder the day before
Work Date	Date	Next planned working day; moves freely up to the Deadline
Target; Achievement	Numbers	Sub-tasks planned; sub-tasks ticked. Yes-or-no tasks: Target 1
Task progress	Calculated	Achievement ÷ Target
Upstream / downstream tasks	Two-way links	Drive both Blocked statuses; roll up share of upstream tasks complete
Related goal(s); origin	Link; text	Which goals it serves; where the task came from
Days left, deadline week	Calculated	From the Deadline; drive the views
PROCEDURE		
Name, ID; Owner	Text; auto ID; person	Detailed enough for someone with no context to get the result
Execution frequency	Select	Daily · Weekly · Monthly · Quarterly · Yearly · Ad hoc
Status	Select	Active · Dormant · Superseded · Retired
Goals impacted	Two-way link	Feeds the support flag on each goal

Any tool that can hold two status sets, two-way links and a few formulas can run this system

Many teams will configure a tool they already pay for rather than build one. Check the tool against six capabilities. Where one is missing, there is usually a workaround. Where the workaround is "a column in a spreadsheet someone updates by hand", the system will decay.

EXHIBIT 13

Six capabilities the tool must have, and what to do when it lacks one

CAPABILITY	WHY THE SYSTEM NEEDS IT	IF THE TOOL CAN'T DO IT
Separate status sets per record type	Goals and tasks need different vocabularies; sharing one list is the failure this system exists to fix	Use separate projects or lists for goals and tasks rather than one list with a type field
Status groups	Roll-ups and views filter by To-do, In progress and Complete, not by each name	Add a group field derived from status by rule or automation
Two-way links between records	Support flags, Blocked statuses and linked-task progress all read the links	Store links on the task; compute the goal side with a saved query
Formulas and roll-ups	Goal Achieved, Linearity, task progress and the support flag	Compute them in a scheduled script or sheet that writes back daily
Field rules and edit history	One owner; dates inside the quarter; a Deadline that never moves	Lock the Deadline after creation, or audit changes to it in the weekly review
Saved personal views and calendars	The four views in Exhibit 11, each filtered to "owner is me"	Saved filters per person; a shared dashboard page as the default home

IF YOU ARE BUILDING IT

1. **Make the metric type a field** (cumulative, or level/average) so Linearity knows which rule to apply.
2. **The 90 in Linearity assumes the goal runs the whole quarter.** If goals can start mid-quarter, store a start date and use the goal's own window.
3. **Reject a goal where Hundred equals Zero.** Both calculations divide by the difference.
4. **Count Abandoned tasks separately in the linked-task roll-up.** Abandoned sits in the Complete group, so a plain roll-up rises when work is dropped.

Build order

- **Goals and tasks,** with their status sets and groups.
- **Links:** goal to task, task to task, goal to goal.
- **Calculations:** Goal Achieved, Linearity, progress, support flag.
- **The four views,** and a dashboard as everyone's home page.
- **Templates:** goal page, task page (page 15), weekly review.
- **A procedures register.** It can start as a simple list.

Roll-out

- **First session:** turn current priorities into goals and tasks, using the control test on every item.
- **Week one:** run the first weekly review, even if every goal is "Not started".
- **First quarter close:** call every verdict, roll nothing over, and open next quarter's goals fresh.
- **After that:** each task that recurs is a procedure waiting to be written.

Two page templates make every goal and task get written up the same way

A|P uses these two templates in client projects. They are a starting point, not a standard: change the headings to suit your team. What matters is that the bet and the definition of done are written before the work starts, not reconstructed after it.

TEMPLATE · GOAL PAGE — SUGGESTED BY A|P

THE BET

If we **[run these procedures / do this]**, then **[outcome]** by the due date, because **[why we think the world will cooperate]**. Serves: **[downstream goal or business outcome]**.

MEASURE

Metric: the one number that counts

Source of truth: where the number comes from

Zero / Hundred, and why: status quo; a fantastic result

ENGINE

Procedures: which ones move this number, and how

Procedure gap: the part of the work with no procedure yet

Tasks: one-offs and experiments only

WORLD DEPENDENCY

Who has to act: reply, buy, sign, rank, approve

Signals to watch: early indicators that set Confidence

VERDICTS, DECIDED NOW

Earned: what must have happened · **Lost:** what "the world said no" looks like · **Forfeit:** what not running it looks like

CONFIDENCE LOG

date — Confidence — evidence — next move (one line per weekly review, newest first)

CLOSE-OUT, AT THE DUE DATE

Verdict · what we learned about the bet · next quarter: drop it, open a new goal, or turn the work into a procedure

TEMPLATE · TASK PAGE — SUGGESTED BY A|P

WHAT AND WHY

What are you trying to do:

Why are you trying to do it:

DEFINITION OF DONE

Closed Fully: all sub-tasks complete

Closed w/Qualifiers: all Must Do sub-tasks complete

SUB-TASKS

Must Do
- ☐ sub-task
- ☐ sub-task

Should Do
- ☐ sub-task

Target = number of sub-tasks · Achievement = number ticked

LINKS

Related goal(s) · upstream and downstream tasks · type: strategic or tactical

SUPPORTING FILES AND NOTES

UPDATES LOG

Quick updates go in comments; anything that needs analysis goes here.

WHAT THIS ADDS UP TO

Twelve checks show whether your current work manager can tell a bad bet from undone work

EXHIBIT 14

The work-manager audit: seven usability criteria and five structural rules

Run against the tool your team uses today; each row passes or fails

	CHECK	PASSES WHEN
USABILITY		
1	Time is not wasted on the tool itself	Updating it is part of the work and the weekly review, not a separate chore
2	Communication is helped	Anyone can see the state of a piece of work without asking its owner
3	Tasks are easy to manage	Creating, re-dating and closing a task takes seconds
4	Related work sits in one place	One source of truth per work stream; no confusion over which list is current
5	No alternate systems crop up	No side spreadsheets, chat threads or notebooks tracking the same work
6	Usable across functions	Product, sales and operations use the same tracker
7	Works for one person or twenty	The same structure serves a founder alone and a 20-member team
STRUCTURE		
8	Goals and tasks are tracked separately	Two record types with two vocabularies: Confidence and Status
9	Every goal has a baseline and a target	Zero and Hundred are set before work starts
10	Verdicts separate skill from luck	Earned and Lucky are different outcomes, and so are Lost and Forfeit
11	Nothing rolls over	Every goal and task is closed inside its quarter
12	Unsupported goals are visible	Goals with no procedure, or nothing at all, are flagged automatically

The pattern

A tool that records activity rather than bets can't show whether a plan failed or was never run. That gap is what this system closes, for a founder working alone or a team of twenty.

ABOUT A|P

A growth partner for healthcare businesses

A|P works with independent private hospitals and healthtech founders in India on the problems that sit between departments: growth strategy, patient acquisition, operations, unit economics and go-to-market.

The approach is bespoke rather than a playbook. Every engagement starts with diagnosing how this particular business works and is measured on outcomes, not deliverables. The system in this report is the one A|P runs on, and the one it sets up for clients.

Aviral Prakash is the founder and principal consultant. He holds an MBA from IIM Ahmedabad and a B.Com (Honours) from Shri Ram College of Commerce. He spent more than a decade in growth, pricing and go-to-market roles at Mahindra & Mahindra, Airtel and Enphase Energy, and today works with healthtech companies on growth and go-to-market.

NEXT STEP

Want to know whether last quarter's misses were bad bets or work that never ran?

Book a 60-minute discovery call. We'll look at how your team tracks work today, and what it would take to have this running before your next quarter starts.

Book a discovery call · Services for healthtech startups

Notes and sources

1 A|P, KNB-48 *What is "Work"* and PNP-10 *Work Management*: the task–goal distinction and the seven usability criteria (Exhibit 14).

2 A|P, PRO-7 *Managing Goals and Tasks*: the quarter rule, the two vocabularies, drift, and how tasks, procedures and goals relate.

3 A|P Goals Tracker, Tasks Tracker and Procedures databases (Notion, October 2026): fields, status groups, formulas, views and the goal and task page templates.

4 A|P, *Instruction Manual — Work Management System*: how to use each tracker, and the success metric for each.

5 A|P, SOL-9 *Work Management System*: roll-out sequence and first session.

Build recommendations: the metric-type field and the "If you are building it" notes (page 14) and Exhibit 13 are A|P recommendations for teams building their own version.

Examples: all company scenarios, goals and figures in the exhibits and boxes are illustrative.