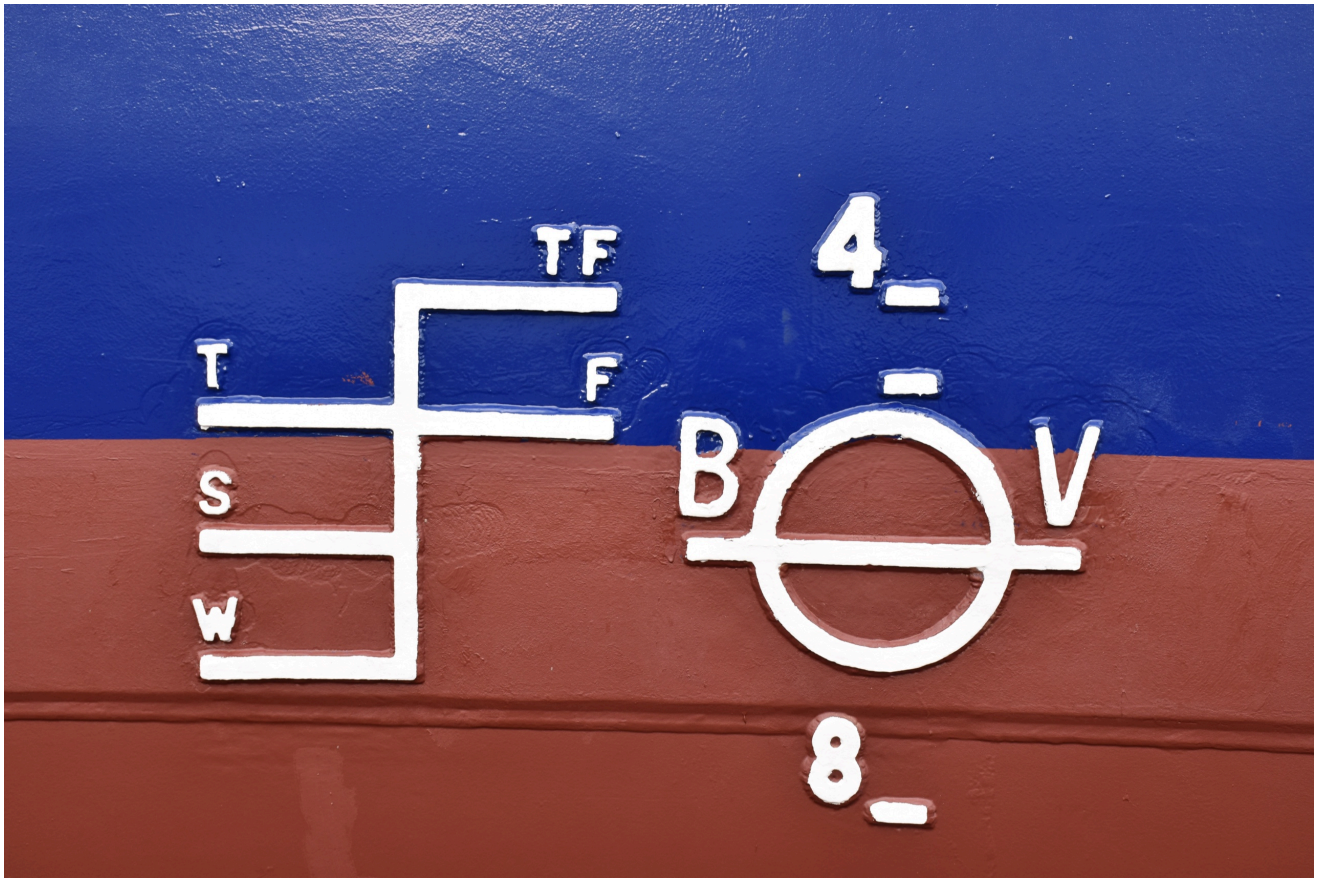
The Load Line Method

Getting the most out of agent assistance while holding quality and control in knowledge work.

Mathew Hager · mathewhager.com · Version 1.7 · September 2026 · loadlinemethod.com



International load-line mark, photograph by Niels Johannes (2025), CC BY-SA 4.0, via Wikimedia Commons [39].

More and more often, we hand the execution of work to agents. That puts the planning, the review and the judgment on us. It is less work of one kind and more of another, and the two are not managed the same way.

Say it is Thursday, and three pieces of work are due tomorrow. One is a proposal for a client. You give an agent your rate card and the scope, and ask it for the pricing table. You do not let it send anything. You point a second agent at two of this month's ledgers and ask whether

they agree. A third drafts a market summary for the board, citing a source for each figure. Then you go into meetings, and each agent is done in minutes.

When you come out at half past four, all three are waiting for you. Now the questions are yours, all at once. The table is laid out and totalled. Is every rate the one on the card? Do the totals follow from those rates? Should this client get these prices at all? Once the table is sent, a wrong rate cannot be un-quoted. The agent's check shows the ledgers agree to the penny, and anyone can run it again. Were those the right two ledgers? The summary reads well. Does each figure appear in the source it cites? You have an hour before you leave, and the table needs all of it. The other two get a glance.

For most of our working lives, writing the thing, building it, drafting it, was the work. That is much less true now. What takes your time instead is setting the work up, preparing what an agent needs, and deciding whether what comes back is right. You also have to decide how much you control and how much you let the agent decide.

That shift changes how you manage work, and how you manage time.

It does not change what you are answerable for. You did not write it. You still own it. Much of what is going wrong with AI at work right now sits in that gap.

The Load Line Method is a framework for working inside that shift.

The name comes from ships. A load line is a mark painted on a ship's hull, and it shows how deep she may be loaded. Put more cargo aboard and she sits lower, until the mark reaches the water. The reading is the gap between the two, and it says how much more she can take.

The Method's job is to tell you how much you can carry and still be any good at it, how to know when you are near the line, what to do when you reach it, and what you will accept when the work comes back.

I served as a nuclear trained and qualified submarine officer, and I hold an MBA. On board I was Assistant Operations Officer, overseeing the teams for communications, navigation, IT and electrical distribution, and for three years the cybersecurity manager. A lot of this paper comes from how we handled safety on board. Since leaving I have run the AI adoption problem twice, in a public company's regulated healthcare business and in a seed-stage startup. In healthcare that meant automated review of submissions to a federal regulator, on data de-identified before any model saw it.

Four definitions

An agent. Any AI helper you hand a task to and get finished work back from. That covers an assistant drafting a memo, a coding agent opening a branch, a research tool reading sources for you, and a scheduled job nobody watches. The breadth is deliberate. What matters here is that work goes out and finished work comes back.

The formal version is Parasuraman and Riley's, and their subject is automation.

> A device or system that accomplishes, partially or fully, a function that was previously, or conceivably could be, carried out, partially or fully, by a human operator [1].

A test. Anything that holds the right answer independently of your opinion, and tells you plainly whether the work matches it. In the proposal, whether the totals follow from the rates is a test. Whether this client should get these prices is not. Nothing holds that answer except your judgment.

Evidence. A fact about the finished work that someone else could verify, and that tells you whether to accept it. It is about the work, never about who or what produced it. For the proposal in the example above, it is every rate traced to your rate card and every total reworked from those rates. A test is one way to produce it. That the agent is usually right is not evidence.

An open decision. One piece of work waiting on a named person to rule on it. Offered, not started and not generated. That person may be you, or someone you named.

What this paper is

Who this is for. You use AI in your day-to-day work already, or you are about to start.

What you get. A method for thinking about how AI changes the way work is done, and the consequences of that shift. It gives you four things to do, starting this week: (1) sort the work a test can settle from the work only you can judge, (2) set how much authority the agent gets before it starts, (3) decide in advance what evidence you will accept when the work comes back, and (4) measure what you can actually carry. Every one of them is a decision you make, not a feature you install. Doing them right still takes building, and it cannot be a paper exercise.

How it is meant to be run, which matters more than it sounds. Read end to end, this looks like a lot of process. A boundary sheet. An evidence list per class. Tripwires. An exception log with expiry dates. A block log. A retrospective every day and again on Friday. The goal here is to understand the concepts, then use AI to carry that process for you.

Where it comes from. None of this is a new kind of problem. It is an old one arriving somewhere new.

Aviation met it when autopilots got good enough that pilots stopped watching them closely. Medicine met it when prescribing software began raising more alerts than anyone could read. Manufacturing met it when the line got fast enough that nobody could inspect every part. Each of these fields learned the same lesson in the same order, and each paid the price before learning it.

Summary

The thing that is often forgotten when you use AI is that you are still responsible for the work you produce. Naming the AI is not a defense—when I hear it, my trust in the work breaks—and ownership does not thin out as the volume rises. The expectation now is that everyone delegates work to agents, or soon will. So anyone using AI is a manager now, whether or not they have direct reports. Managing well is not a new problem. There is a deep literature on it, and adapting that literature to agents is the work now.

What is left in human hands is planning, review and judgment, and that is where the load has moved. Yours, and your team's if you have one.

Not all of that load is real, because review runs on a spectrum. What it costs to be wrong decides where on it you sit. Some work you can accept without looking at all. Where a test can settle the answer, send it to an agent and let it run, because your reading of it adds nothing. You still own the result, even when you do not do the work.

For everything else there are two decisions, made at different times. Before the work, risk sets how much authority it gets. Judge it two ways: how likely is it to go wrong, and how badly would it hurt if it did? Then after the work, a second decision. You accept on evidence you named before it went out, and on nothing else.

It also helps to not treat a task as a single decision. Gathering, analysing, deciding and acting are four, and each can be given a different amount of authority. So an agent can

gather with full authority and still not be allowed to act. Figure 3, in section 2.2, prints the stages and the scale.

None of that makes review reliable, because checking after the fact cannot catch everything. That is provable rather than pessimistic. Protection comes from the controls you set before the work starts rather than leaning on inspection at the end.

Knowing how much you are carrying needs an instrument, and this paper works through the options for tracking the load you and your team hold.

Your own mark works like the ship's. The limit in this version is borrowed from published work rather than measured here. A start rule clears agent work to begin, keyed to your capacity, workload and fatigue rather than to how good your agents are. The mark itself has to be measured. Until you have done that, every number in this paper is somebody else's.

Time is the other half, and agents did nothing for it. They made producing cheap. They did not make your week longer. Every piece of work you take on displaces something else.

Sort the work deliberately and you carry far more than you could manage by reading all of it yourself. Sort it blindly and you carry less than you managed before the agents arrived. You end up reading everything and slowing down, or reading nothing and shipping work nobody checked.

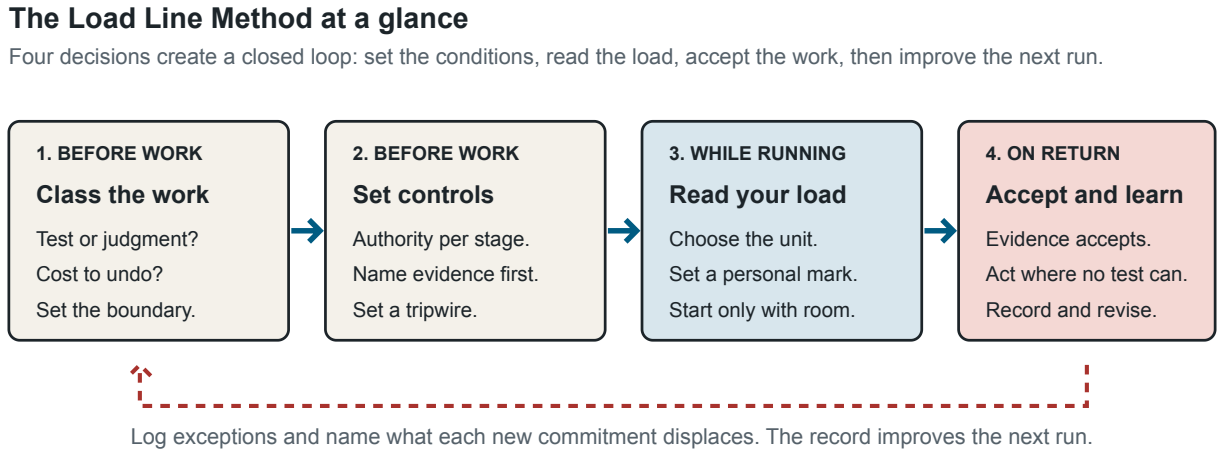


Figure 1: the Method at a glance. Class the work, set the controls before it starts, read the load, then accept and learn from what returns.

1. Where the work now lands

1.1 What you own

You send an agent a task, and what comes back reads well. You skim it and pass it on. Who owns it now?

You do. Whoever hands the work on owns it, however it was made. You own whether it is accurate, whether it fits, and whether it is worth anything. You are the author. Read it and improve it before anyone else reviews it. Do not name the AI as cover. Hedging like that is a sign the work may not be ready to share. Software engineering has begun writing that split into practice. GitLab's public handbook states it plainly [2]: automate what an agent can do, and put human effort where judgment is essential.

That is a judgment call, and someone bothered to write it down. Most people reading this have made the same call and never written it down.

Written down, it is a rule. Anyone can follow it and anyone can hold you to it. Left in your head it is a preference, and the people around you have to guess. They will guess differently from each other, and differently from you.

Ownership does not thin out as the volume rises. Ten pieces of work a day are owned the way one was. Work can be shared. Ownership cannot.

Rickover put that to the Joint Committee on Atomic Energy in 1961 [3]. Responsibility resides in a single individual. You may delegate it, but it is still with you.

That sentence gets read as a license. I am still responsible, so I may take on any amount. His own program kills that reading. His headquarters ran the crew examinations itself when he said it. Three years later it could not keep up with the growing fleet. He did not work harder, and he did not quietly lower the standard. He stopped doing the examining and kept appointing the examiners and reading their reports [4].

So the steadiness of ownership is what lets you take on more, and it names no amount. Running more work is a question of how, never of whether. The man who wrote the sentence hit his own ceiling and rebuilt the mechanism underneath it.

Individual contributors now need an operating discipline good managers already built. Young managers learn those lessons the hard way: unclear direction, missing guardrails, vague

expectations about what finished means. The same lessons now bite anyone running agents, whatever the title says.

There is a plainer reason to care. Senior people have always had somebody standing between them and the work. A chief of staff, an assistant, an editor, deciding what deserved their attention and turning the rest away. That was never generosity. It was arithmetic. Screening costs a salary, so it went to the people whose attention cost more than the salary did. Machine help changes that arithmetic, and the same protection now reaches everybody else.

1.2 What you can take in

Execution moved to the agents, and it moved wholesale. Planning, review and judgment moved too, but only partly. That shift changes what a work day is made of.

Producing is cheap. Reading what comes back is the job. Assessing it, judging it, giving feedback that changes the next effort, and communicating in both directions.

How much you can carry and stay effective depends on what you can take in. The load sits in a head, one reviewer at a time.

Underneath that limit is an information-theory theorem showing why your review capacity cannot be stretched indefinitely.

Call anything that holds something else in bounds a regulator. A thermostat is one. So is a reviewer.

The theorem says a regulator needs at least as many distinct responses as the thing it regulates has distinct states. Say the work can come back wrong in twenty ways, and you can tell four of those ways apart. The other sixteen go out unregulated. They do not disappear. They show up in the result.

Ross Ashby proved that in 1956 and called it the law of requisite variety [5]. Variety is his word for the count of states you can tell apart.

In review work, you can only catch errors up to your capacity to process information. As Ashby proved, a regulator can be no better as a regulator than it is as an information channel [6]. You can regulate what you can take in. Up to that capacity, the work ships on your judgment. Past it, it ships on the agent's.*

The extension to people is his, not mine. In 1958 he applied it to the surgeon, and to the manager whose business prospers or fails [7]. A person's intelligence, he wrote, cannot exceed their capacity as a transducer, measured per second or per question.

That leaves you with a limit and no number. The theorem never says a person can supervise five agents, or fifty. It says the limit is there, that yours depends on the distinctions you can draw, and that the person next to you has a different one.

The objection to borrowing a theorem like this into management is in print [8]. It lands the moment I use it for a number, so I do not. Ashby allowed the loose fit himself, on the grounds that only the existence of the limit is at stake.

Nobody can hand you your number. You have to find it yourself.

There is a second proof in the same book [9]. Anything that corrects only the errors it can see will never catch them all. Reviewing work after it exists is exactly that. So some of it gets past you, every time, however carefully you read. That is the shape of the loop, and it has nothing to do with how tired you are.

1.3 What review costs

What is left to you is review and judgment. Push more work through your reviewing capacity and your catch rate erodes. Three things are worth knowing. They are measurements rather than proofs, so they carry a different kind of weight from the theorems above.

First, the uncomfortable one: steady reliability is what erodes your catch rate. Poor reliability does not.

Parasuraman, Molloy and Singh ran forty people through a flight simulation [10]. When the machine's reliability was held constant, they caught about a third of the planted faults. When it varied, they caught more than four fifths. The task load was identical. When watching was the only job they had, they caught almost all of them. Bagheri and Jamieson confirmed the effect [11], and an extra hour of practice did not reduce the drop [12].

Carry that across and it is uncomfortable: the better your agents behave, the fewer mistakes you catch. The effect is worst when you are running several things at once and mildest when watching is all you are doing, which is the setup the Method describes.

That is the argument for running fewer things at a time. Section 3.1 is where the paper puts a number on that, and the number is borrowed rather than measured. Nobody in those studies supervised a generative system, so the transfer is my analogy.

Second, you cannot feel it happening. People held on restricted sleep were largely unaware of their own decline. At their worst they reported feeling only slightly sleepy. That was measured on reaction time rather than judgment [13]. Sustained watching does something similar on a much shorter clock. Across twenty minutes, lapses rise and the threshold drifts toward whatever the default answer is [14]. Here the default answer is yes. The drift looks like approving without reading. What moves is the bar you are applying, not your ability to tell one thing from another.

Third, load and fatigue each cost you. A unit rise in workload made a person 1.75 times less likely to catch what the machine missed. A unit rise in fatigue, 1.40 times. How much authority the machine held moderated neither [15].

Then the objection people put first. A second person is supposed to stop bad work before it ships. In practice, human second checks regularly fail to catch errors.

When an unstable landing approach calls for a go-around, the procedure is to abandon the landing and come round again. Pilots fly the go-around in about 3 percent of the cases that call for it [16]. The pilots' own managers guessed compliance was near 20 percent. Medicine shows the same thing in software. Clinicians override drug safety alerts in half to nearly all cases [17]. Most of those overrides are correct, which is exactly why the alert stops being read. Making an alert nearly impossible to dismiss raised compliance from 13.5 percent to 57.2, but that trial was stopped early after delayed treatment harmed four patients [18].

Selective filtering works better than placing a second pair of eyes on everything. A Swedish screening trial let a risk score decide which mammograms went to a second human reader [19]. Detection held, and reading volume fell by 44 percent. The instrument routed the cases that warranted scrutiny; it replaced nobody.

How big the degradation is remains contested. Two direct tests failed to reproduce it [20] [21], and the meta-analysis behind it separates into two findings [22]: routine performance rises as automation does more, but performance when taking control back falls. That second half is what should worry you.

Two of the objections bite on the Method directly, so they are printed here rather than buried. Readiness you assess in yourself failed as a measure on 475 people, so any instrument that asks how sharp you feel today is weak [23]. And without a stated benchmark, calling somebody complacent is a verdict rather than a measurement [24]. Both are aimed at the start rule section 3.2 builds, and neither has been answered.

What survives is narrow, and it is enough to build on. Catching degrades under steady reliability, and under load and fatigue. You cannot feel either happening. That is what you are protecting while you get more effective.

1.4 What never reaches you

So far this section has said one thing. Your attention is the limit, the reviewing you do with it wears down, and you cannot feel that happening.

That claim has a boundary, and the boundary is wide. Where a test can tell you the answer, that limit does not bind.

Much of the work is like that. The build compiles or it does not. The suite is green or it is red. The totals reconcile and the links resolve. The spelling is checked by something that never gets bored. In all of it something other than your judgment holds the end state and can rule on it.

There you can load as hard as the checks will take. Judging is cheap, so your own limit never binds. The line runs on verifiability, not on domain.

Drawing that line is yours. Nothing in the Method sorts your work into the two kinds, and no instrument in it can tell you which pieces a test could settle. You do that by hand, one class of work at a time, and section 2.1 is where you write it down.

2. How to carry more of it

Section 1 leaves you with three facts and no instructions.

You own more work than you can read line by line. The reviewing that covers the rest wears down without telling you it is wearing down. And a large part of the work never needed reviewing in the first place.

Two questions follow: how much can you carry and still be effective at it, and what will you accept when the work comes back?

2.1 What you can send away

Some of your work never needs your attention at all. This subsection is how you find it, and how you find the work at the other extreme, where your judgment is the only thing standing between a mistake and the customer or colleague who has to deal with the consequences.

Two questions sort it, and they are independent of each other.

Can a test settle it? Where something other than your own judgment holds the end state, load that work as hard as the checks will take. Send it to an agent and let it run. Do not review it by hand because reviewing feels responsible.

What does being wrong cost to undo? Not what it costs to correct the document. What it costs to undo the damage once the mistake has left your hands and nobody caught it.

The cost to fix an error and the cost to undo its consequences are usually nothing alike. A rate in a client proposal takes ten seconds to edit and cannot be un-quoted once it has been sent. A number in a board pack takes one keystroke to fix and cannot be un-decided once the board has voted on it. A misread clause in a contract costs nothing to retype and everything to litigate.

The cost you are weighing is the second one, across the whole system the work touches and everyone downstream who can act on it.

Arrow and Fisher proved the useful form of this in 1974 [25], and it is more forgiving than it sounds. An irreversible choice is not forbidden. It simply carries an option cost: committing now forfeits the value of information you might learn by waiting. If waiting a week costs $2,000 in delayed revenue but could reveal information worth $5,000, waiting is the cheaper choice. Irreversibility makes a choice cost more; it does not put it out of reach.

Their condition is the part usually dropped. **The discount exists only if you will know more before you would have to act again.** No learning coming, no discount. Irreversible and nothing to learn is just a decision.

The two axes cross, and the crossings are where readers get stuck.

Classify the decision before the work begins.

Ask what can settle it, then what it would cost to undo an error after it has left your hands.

	A test can settle it A check holds the answer.	**Requires your judgment** No artifact proves the act.
Cheap to undo The harm can be reversed.	**Send it away.** Accept the test's verdict. Your reading adds nothing.	**Outside the boundary.** You still author what goes out. Low stakes do not remove ownership.
Costly to undo The harm travels downstream.	**Inside the boundary.** The agent can produce evidence. Proving it is cheap.	**Inside the boundary.** The agent prepares the work. You perform the final act.

Draw the boundary once per class of work, not when you are tired and want it finished.

Figure 2: classify the decision before the work begins. The bottom-right square is where the final act remains yours.

That bottom right box is the one the Method exists for. Sending a client email. Publishing a page. Paying an invoice. Signing off a hiring decision. All costly to undo, and no test anywhere holds the right answer.

Nothing outside the boundary is accepted at face value unless an automated test settled it.

Write the boundary down before the work. One page.

Work in classes rather than one piece at a time. A class is a kind of work you do repeatedly and can name in a few words. Client-facing pricing. Published copy. Anything that moves money. Board reporting. You decide once for the class, and every piece inside it inherits that decision, which is what stops you re-deciding while tired.

The sheet lists the classes whose consequences you cannot undo, and beside each the evidence that accepts them. Keep it somewhere the work can see it rather than in your head. It wants to be the kind of artifact a tool can read, so that anything checking your work can check it against the sheet rather than against your memory of the sheet.

Controls this rigid are meant to be rare. Applying them everywhere is not caution, it is a failure to manage risk, and it costs you the capacity the whole Method exists to give you

back. The point of drawing a boundary is that most of your work sits outside it and moves fast. What matters is that the few classes inside it are treated without compromise, and that you decided which ones those are before you are tired.

The Navy's version of that line is the strictest one I know, which is why it is worth seeing.

It has a name, SUBSAFE, and it exists because of a loss. The submarine Thresher went down in April 1963 with 129 people aboard. Ultrasonic testing had checked 145 of roughly 3,000 silver-brazed pipe joints exposed to full submergence pressure, on a schedule that would not delay the ship. Fourteen percent of that sample failed. Extrapolated across the boat, several hundred joints may have been substandard [26]. The program that would have caught them did not exist. Two months later it did.

We did this in the submarine force with a Certification Boundary Book, which is where SUBSAFE starts. There was one per class of submarine, drawn before anything was built, and it named which systems were inside the boundary and which were not.

What made it work was how narrow it was. It did not cover the submarine. It covered watertight integrity and the ability to recover from flooding, and nothing else. Whether the boat could complete its mission was a different program's problem. Keeping the certification boundary narrow is the trade that bought zero tolerance: because it covered only what was truly fatal, the program could afford to be uncompromising about everything left inside.

The point is that your boundary line is written down in advance, as an artifact, rather than being a judgment you make at five o'clock on a Friday about work you want to be finished with.

Two cautions keep the exclusion honest.

First, a passing test says the work matches the test. It does not say the test was the right one to write.

Go back to the two ledgers from the opening. The agent writes a check, runs it, and the check passes. Every row in the first ledger appears in the second, to the penny. The proof is real and you can rerun it yourself.

Now notice what nobody checked. Whether those were the right two ledgers. The reconciliation cannot tell you that, because it was never asked. It compared what it was

pointed at. Choosing what to point it at is the part that stayed with you, and no test you can write will do that part, because writing the test is that part.

So the boundary usually runs through a job rather than around it. The agent proves the arithmetic. You own the question.

Second, a run of good behavior is not a license. How well the agent has behaved lately is not a reason to stop checking.

This is the hardest rule in the paper to follow, because it runs against the thing your judgment is built to do. You learn from experience. Twenty clean runs is exactly the evidence a sensible person uses to relax, and relaxing is exactly what the measurements say destroys your catch rate. Steady reliability is what erodes your catch rate. Poor reliability keeps you sharp.

So the signal you would naturally read is the one that is inverted. A run of good behavior tells you the agent is doing well and tells you nothing at all about whether you would notice if it stopped. Read the evidence instead, every time, on the classes where being wrong is expensive.

Sorting well is the skill. The sheet is what makes you do it once, in daylight, instead of forty times under deadline. It is also the thing a colleague can argue with. A line you keep in your head is a line only you can audit.

2.2 Two decisions, and when you make each

Steady reliability erodes your reviewing, and you cannot feel it happening. Nothing proposed here repairs that.

Two things can still be done about a risk you cannot remove. You can reduce how much rides on your reviewing at all, which is what the rest of this subsection is for. And you can decide in advance what you will do when a named condition is met, so the decision is not being made by whoever you are at the time.

Call those tripwires. A tripwire names a condition and the response to it, both written before the condition arrives. Open work goes past your number, so nothing new starts until something closes. A class of work you had outside the boundary produces a near miss, so it moves inside. A second exception is logged against one class in a single month, so that

class gets re-cut rather than re-excused. None of that needs judgment in the moment, which is the point.

So this subsection asks two questions rather than one. How much authority does a given piece of work get, and what will you accept when it comes back?

Those are separate decisions and they happen at different times. The first runs before the work and buys authority. The second runs after and buys acceptance. Making both at once, on the way out, is the error this section exists to fix.

I ran this for years in the submarine force, and of everything that program did, this is the part I would take anywhere.

Keeping the two decisions apart is what stops a deadline from rewriting your standards. Decide how careful to be while the work is still abstract and nothing is late. Decide whether to accept it against a list you wrote when you had nothing to gain from a short list. Outside that program, almost nobody separates them, and both calls end up being made at the same moment by someone who wants the work finished.

The measured case for splitting them. The 2024 DORA report found rising AI adoption associated with a rise in delivery instability, and the 2025 report found the relationship persists [27]. Instability is their word, and it is the one to keep. Output was not what degraded first. The ability to recover was.

Decision one: how much authority. Two things settle it. How likely the work is to go wrong, and how bad it is if it does.

Parasuraman, Sheridan and Wickens put it in one line [28]. Risk is the cost of an error multiplied by the probability of that error.

Multiply a cost by a probability whenever you can estimate both, because both are real quantities and their product means something.

Where you cannot put numbers on it, say so and use judgment rather than dressing the judgment up as arithmetic. Multiplying labels together, high by moderate, is not a calculation, and ranking risks that way can put the smaller one above the larger [29].

I prefer to look at it a second way, and the Navy supplies it. Operational Risk Management is the service-wide doctrine for exactly this call [30]. What is interesting is that the Navy runs it

alongside SUBSAFE rather than instead of it. Two different answers to risk, inside one organization, doing two different jobs.

ORM scores a hazard twice. Severity, in four categories. Probability, in five. It combines them into one alpha-numeric code, and three things about that code are why it is worth borrowing.

It is not a product. The two are put side by side rather than multiplied, and the grid that displays them is optional in the instruction's own words.

It does not decide anything. The instruction calls the ranking a guide to relative priority rather than an order to follow, and warns in writing that a hazard's worst outcome may not carry its highest code. No score anywhere in it triggers an automatic accept or reject.

The accept call belongs to a named person, who weighs what risk is left against what the work is worth.

So ORM splits the same two decisions this subsection splits, which is the confirmation worth having. What differs is what governs the second one. SUBSAFE requires verifiable evidence. ORM requires the right authority.

That difference is the whole problem for one person running agents. Authority you already have, all of it, which is why it cannot govern anything. Evidence is the thing you still have to build. So the scoping comes from ORM and the acceptance comes from SUBSAFE.

Estimating the probability is the harder half, and agents make it harder in a particular way.

Start with what the number would have to mean. A published reliability figure is an average across many runs. Your work is one run, and what you need to know is how far a bad run can fall from that average, which the average does not tell you.

Then the shape of the failure. An agent very rarely produces nonsense you would catch at a glance. It produces something reasonable, built on an assumption that does not hold for your case. It will be well written and internally consistent and wrong at the root. That is the failure you are estimating, and it is the one least likely to look like a failure.

Make this call once per class, not once per task. Change management practice calls a pre-authorized class a standard change, and the point is that the risk assessment is not repeated per instance [31]. Re-deciding every time is how a tired operator drifts.

Low on both earns more authority and fewer checkpoints. High on either earns more context, a shorter cycle and a named trigger for escalation. When risk rises the loop tightens first. Taking the work back is right when the severity is high enough.

Authority is four dials, not one. A task is not one decision you either hand over or keep. Parasuraman, Sheridan and Wickens split it into four stages [28]. Gathering the information. Analysing it. Deciding what to do. Doing it.

Each stage can be set to a different amount of authority, and the settings are independent of one another. So an agent can gather with full authority, analyze with full authority, offer you one course of action, and still not be allowed to act.

On the work that would cost most to get wrong, consider turning the third stage down further still, and asking for the state of things rather than for a recommendation. There is one small study behind this, from 1990 [32]. Thirty-two people in groups of eight, on an identification task. Status alone did better than status with a recommendation attached, and the reading is that a recommendation anchors you before you have looked. Eight people per cell, from 1990, so treat it as a hypothesis with a single test behind it rather than a rule.

The deciding stage has a published scale under it, and the other three do not [33]. It runs ten rungs, from a machine that offers no help at all, through one that proposes and waits for your approval, up to one that decides, acts, and does not tell you.

Set gathering, analysing and acting by description instead, because no scale exists for them. That is a real gap rather than an omission here.

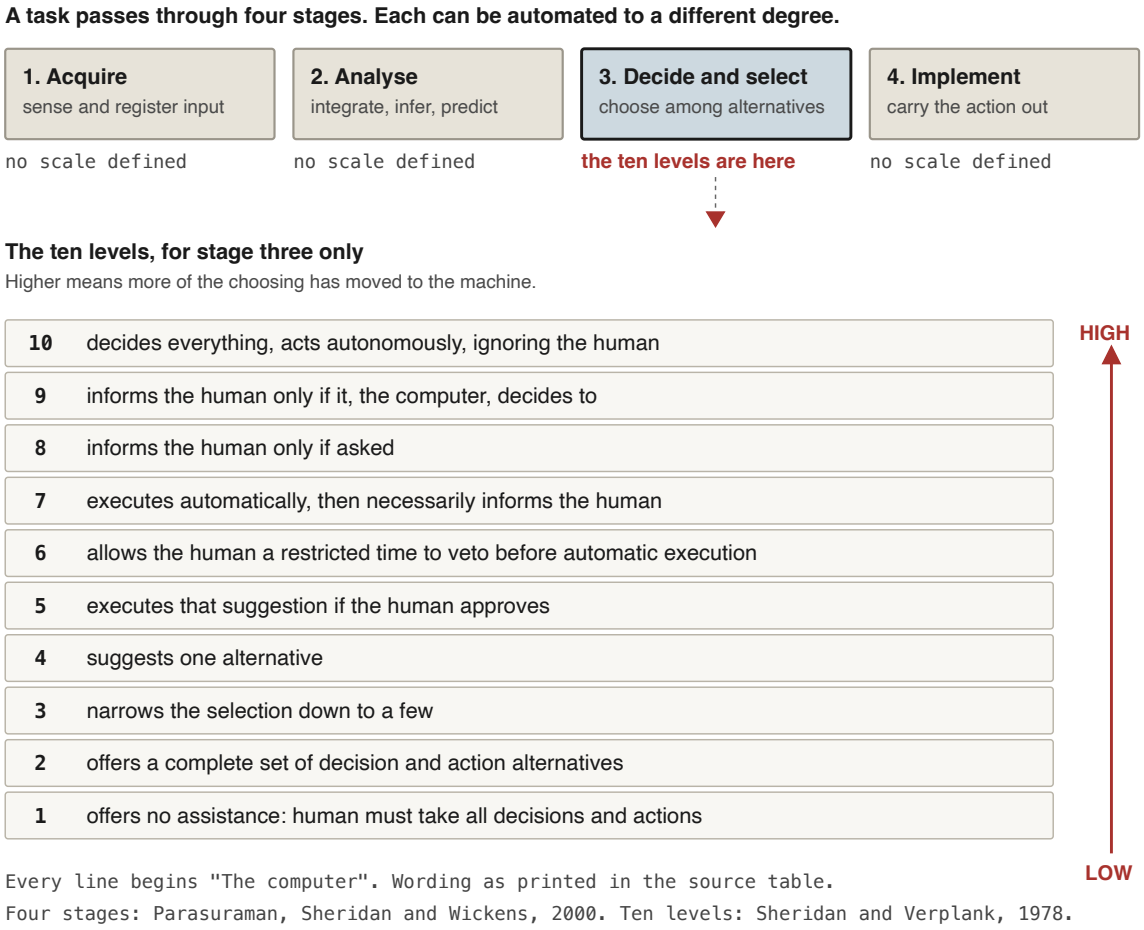


Figure 3: the four stages, and the ten levels defined for deciding. The other three stages have no published scale.

Where a wrong decision is costly, the stage to hold low is the last one. Parasuraman recommends exactly that, and the evidence list below turns it into a rule.

Five things you can do about a risk, not two. So far the answer has been to tighten the loop, and sometimes to take the work back. ORM lists five, in the order people actually reach for them [30]. Reject it. Avoid it by doing the job another way. Delay it. Transfer it to whoever is better placed to absorb it. Compensate for it with redundancy.

Two of those are cheap. **Delay** is available whenever nothing forces speed, and waiting either shrinks the risk or lets a better option appear. **Compensate** means a second agent checking the first, or a draft that goes out in a form that can be corrected before it counts.

Refusing to delegate low-risk work is its own failure. It spends attention where none is needed and starves the work that needs judgment.

Decision two: what you will accept. Name the evidence before the work goes out, and accept on that and nothing else.

SUBSAFE has one sentence at the center of it, given to Congress in 2003 [26].

> Without objective quality evidence there is no basis for certification, no matter who did the work or how well it was done.

Objective quality evidence has a formal definition, and it is broader than it sounds.

> Any statement of fact, quantitative or qualitative, pertaining to the quality of a product or service, based on verifiable observations, measurements or tests.

Two things in that are worth pulling out. Qualitative evidence counts, so this is not a demand that everything become a number. And the evidence attaches to the product, never to whoever made it. That is the half that does the work here, because an agent has no record you could stand on anyway.

Here is the difference in one case, the market summary from the opening. It cites six figures. It reads clean and well written, and it comes from a model that has never given you a bad one. That is reputation, and it accepts nothing.

The evidence is the six sources opened and the six figures found in them. Ten minutes. If you named that before the work went out, you do it and you are finished. If you did not, this is the moment you will decide that the writing is good enough to trust, which is the decision the rule exists to stop you making.

A low reading on decision one does not soften decision two. It moves where the boundary sits. It never changes what it takes to get past it. Low risk buys a shorter evidence list, never a glance at the list you named.

What counts as evidence. It has to be about the artifact, never the producer. The Navy can point at the producer because the welder is qualified in that exact procedure and the instruments sit in a calibration program. An agent holds no such qualification and its behavior changes with each release.

- A test run, with its output, that you or anyone else can rerun.

- A citation that resolves, checked against the source rather than against the claim.
- A number reproduced from the inputs.
- A check by a second agent that did not produce the work.
- For an act nothing can settle, your own hand on the final step.

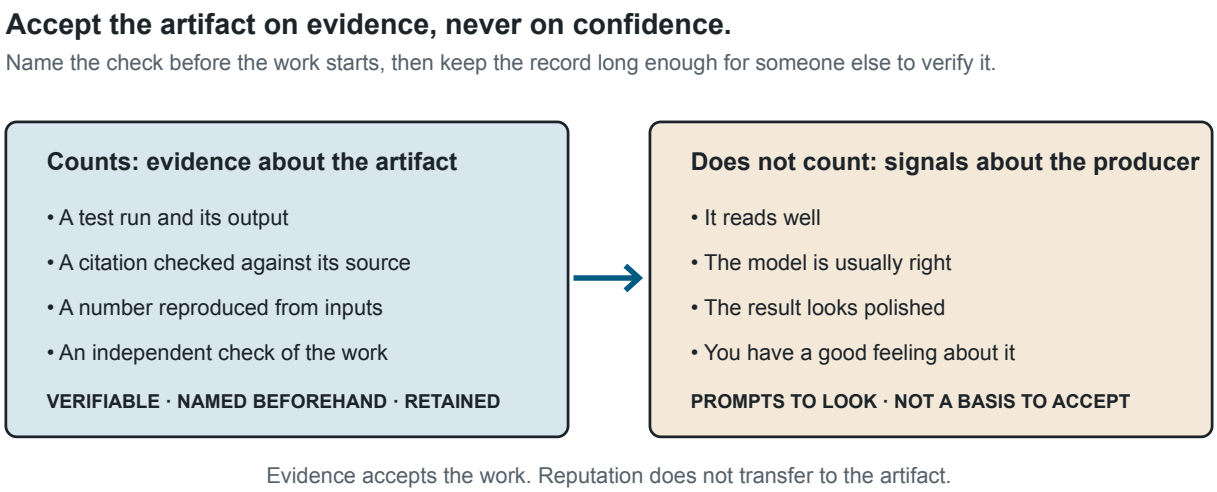


Figure 4: evidence is about the artifact. A result may look good and still need an independent check.

And the record has to outlive the moment. Name the source well enough that anybody can go and look, not just you and not just this week. Say when it held and when it stopped holding. When something turns out to be wrong, close the window on the old belief rather than erasing it, because a record that quietly repairs itself is worth nothing to whoever reads it next. Being replaced is not the same as being wrong.

The last item on that list is the answer to the bottom right box. Where no artifact proves it, the agent prepares and you perform the irreversible act yourself. Parasuraman calls this error trapping, and he recommends keeping the implementing stage low precisely where a wrong decision is costly.

What does not count. That it reads well. That it is well formed, which says nothing at all about whether it is right. That you have a good feeling about it.

The first appendix runs four pieces of work through all of it, one from each box of the matrix in 2.1.

Reopen accepted work and it goes back through. Anything already accepted and handed on is uncertified once changed, until its evidence is produced again.

This one bites harder with agents than it ever did with people. Ask for a small change to something already signed off and you will often get the small change plus three you did not ask for. A tidied heading, a rewritten sentence two pages away, a helpfully updated figure. What comes back is larger than what you asked for, and the parts you did not ask for are the parts nobody is reviewing, because your attention is on the change you wanted.

So the rule attaches to the artifact rather than to the edit. Touch it and its evidence lapses, however small the request was. A draft you are still writing has not been accepted, so nothing lapses.

Sign the record with a name and a date. Signing your own record proves nothing to anybody else. It is there so the date is on it, and so a later reader knows who to ask.

Give every exception an expiry date. Log a line each time you accept without the evidence you named, and write the date it expires. Aviation does this with deferred maintenance, where the category carries the deadline rather than a conversation [34].

The count is the decay signal. At the 2003 hearing the committee chairman put NASA at almost four thousand waivers, a third of them over ten years old. Standing exceptions are how a system rots, and nobody notices until somebody counts.

The run, in order.

Once, before any work.

1. Write the boundary sheet. The classes whose consequences you cannot undo, and the evidence that accepts each.
2. Set the authority each class gets, so you are not re-deciding under pressure.

Every piece of work.

3. Name the class, and copy its evidence into the brief. Do not compose the test here.
4. Set the trigger that brings it back early, and how long it runs before you look.
5. While it runs, hold that trigger and the checkpoints you set. Vague unease without a specific defect is not a checkpoint; do not interrupt the agent just because you feel nervous. Only an actual defect in the work outranks the plan.

6. Accept on the evidence you named, or refuse. A summary of the evidence is not the evidence.
7. If nothing can settle it, perform the final act yourself.
8. Log an exception with an expiry date if you accepted without the evidence.

Monthly.

9. Read the exception log. A growing count, or an expired line still open, is the signal.

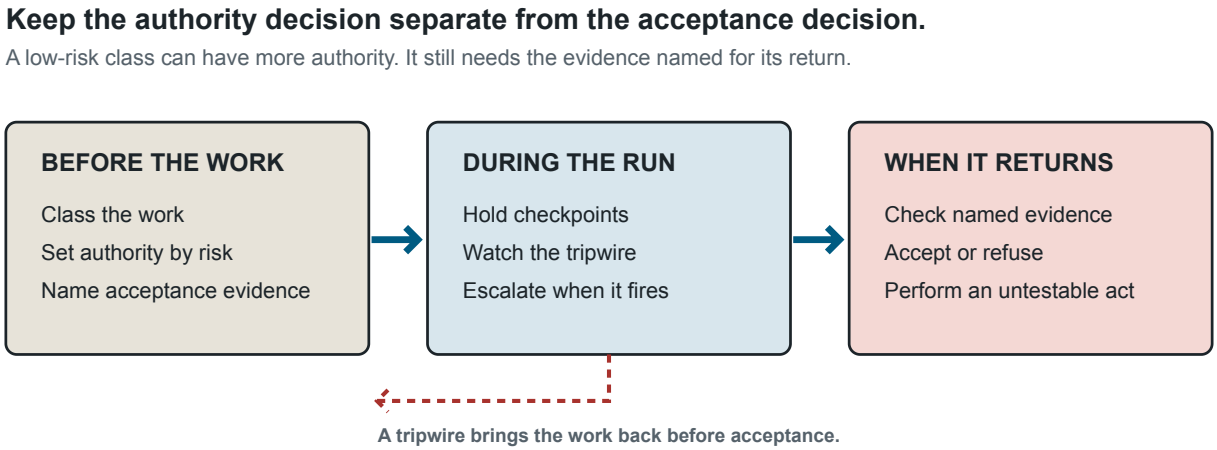


Figure 5: authority is set before the work. Evidence and tripwires govern the run. Acceptance happens only when the named evidence is present.

What one person cannot copy, and it matters. The Navy runs these two decisions across three authorities, held by different people. A program manager, an independent technical authority, and an independent safety and quality authority. Any of them stops the job.

Alone, you are all three, and you are also the one under the deadline. That is the configuration the program exists to prevent. Separating the two decisions in time substitutes for separating them across people, and that substitute is weaker.

There is one more difference. When SUBSAFE leaves something outside its boundary, that thing is not unprotected: fire, weapons, or reactor safety programs catch the rest. When you work alone or on a small team, you have no second safety program underneath yours. Whatever you leave outside your boundary is genuinely outside, and that is worth knowing when you draw it.

So the same class of work sits on different sides of the line depending on what is underneath you. That is the practical instruction. Draw your boundary a little wider than an institution would draw it, and when you are deciding whether a class belongs inside, ask who else would catch it. If the honest answer is nobody, it belongs inside.

Be clear about what the split does not fix. Section 1.2 showed that anything correcting only the errors it can see will never catch them all, and accepting work on evidence is exactly that. Some of it still gets through. No arrangement of two decisions changes the shape of the loop.

Evidence produced by an automated check costs you almost none of your scarce attention. You save your judgment for the work that actually needs it, instead of spending it on work that only needed a verification.

2.3 Finding your unit

Section 1.2 said you can only regulate what you can take in, and that the limit is personal. Applied here, that means the number of agents you can run, and the number of tasks you can have them holding at once, is not set by how many you can start. It is set by how many distinct calls you can still make well in a day. Starting is free now. Deciding is not, and deciding is the part that was always yours.

That bound arrives with no number in it. A number needs a unit and a reading, and this is where the Method produces one.

Start with what gets counted. An open decision is one piece of work waiting on a named person to rule on it. Not started. Not generated. Handed over to somebody for a decision.

Two cases settle the rest. Twenty agents working together on one deliverable is one open decision, because drafting is not handing over and you will rule on it once. One instruction that sets twenty separate pieces of work running is twenty, and they will land on you together.

That second case is why a limit has to bind when the work is launched rather than as each piece arrives. By the time you can count them they are already yours.

The count comes from the work, not from typing. American air traffic control keeps those two jobs apart [35]. A machine predicts how many flights will be in a sector, minute by minute. The limit that number is compared against is set separately, one value per sector. Nobody types the demand in, and nobody argues with it in the moment.

The counter inspects your existing work trackers—boards, pull requests, draft queues, and task lists. It tallies unfinished work without any manual logging: tasks started and not delivered, drafts open and not sent, decisions logged and not made, and jobs queued and not run. It watches the places the work already lives.

On 7 September 2026 it reported twenty-six. Sixteen pieces of work started and not delivered, one draft open, one decision logged and unmade, and eight jobs queued.

Read that number knowing three things about it. It is a floor, because it misses work nobody wrote down. It counts one person's own working state, so point it at a shared task system without scoping it to you and it will count other people's work as yours. And it counts what you are holding on your desk rather than what has been handed over to a reviewer, which is the harder defect. Working alone, a finished piece of work has been put in front of nobody. In a team the same piece has a named reviewer waiting on it, and the reading moves much closer to an actual handover.

Recurring work escapes it entirely, and not by accident. The counter finds work by finding the thing it produced. A project produces something, so it can be counted. The weekly report you have written ninety times produces nothing new to find, is never handed to anybody for a decision, and never starts or finishes in a way a counter can see. It still takes your Tuesday.

Then the harder point. A count is not a load. A count says how many things are open. It says nothing about whether any of them can be finished.

Say two things are open and four hours are left. The hours arrive in ten minute pieces. Nothing finishes. The tally says there is room and there is none. It misleads the other way too, when six of eight open things are waiting on somebody else.

So a limit needs a unit underneath it. The Method names six and lets you pick.

Consolidated block-hours. Most work needs a long stretch of time, and small slices of it are waste. That is Drucker's argument in **The Effective Executive** [36]. This suits work that requires minimum blocks of uninterrupted focus.

Returning review demand. Lead five people on five priorities and the job becomes timing review so you are not the thing everybody waits on. This suits anyone whose calendar is other people's deadlines.

Open decisions waiting on review. That is the automated tally described in Section 2.3. This suits anyone who can derive a number with no typing step.

Unfinished items carrying attention. Unfinished tasks demand mental attention even when you are not actively working on them, including work that is done but never formally closed [37]. This suits load that is felt rather than scheduled.

Effect, rather than activity or hours. Judge by what changed, never by hours logged. GitLab measures its people that way. This suits anyone judged on what changed.

Readiness. Some days the time is there and the head is not. It is the weakest of the six, for the reason section 1 gives about self-assessment.

Choosing is simpler than the list looks. Wait for the next week that goes wrong, then ask what ran out. That is your unit. Name it and read it before any limit binds, because a limit on the wrong unit will read healthy in the week it fails you.

Choose the unit by the constraint that failed in a difficult week.

Use the unit that notices the shortage before it becomes a bad week. Then read it before setting a mark.

What ran out?	Useful unit	Why it helps
Long, uninterrupted time	**Consolidated block-hours**	Work needs a usable stretch.
Other people's deadlines	**Returning review demand**	You are the review bottleneck.
Too many calls await you	**Open decisions offered**	A count can be derived.
Unclosed commitments keep returning	**Unfinished items carrying attention**	The load is felt, not scheduled.
Activity rose but nothing changed	**Effect**	Judge outcomes rather than hours.
The time was there but you were not	**Readiness**	Useful, but weakest on self-report.

Figure 6: pick the unit by the constraint that failed in a difficult week, then read it before setting a limit.

3. Where the mark goes

Sort what a test can settle, set the authority by risk before the work, accept it back on evidence you named, and read your own load in a unit you chose. Every answer so far has been a rule you could follow today. None of them tells you how much you can carry, or how you would know.

3.1 Two marks and the gap between them

A unit still needs a mark to sit against. The load line is what lets a ship load to her safe maximum. It is two marks, not one, and that is the first thing Figure 7 teaches.

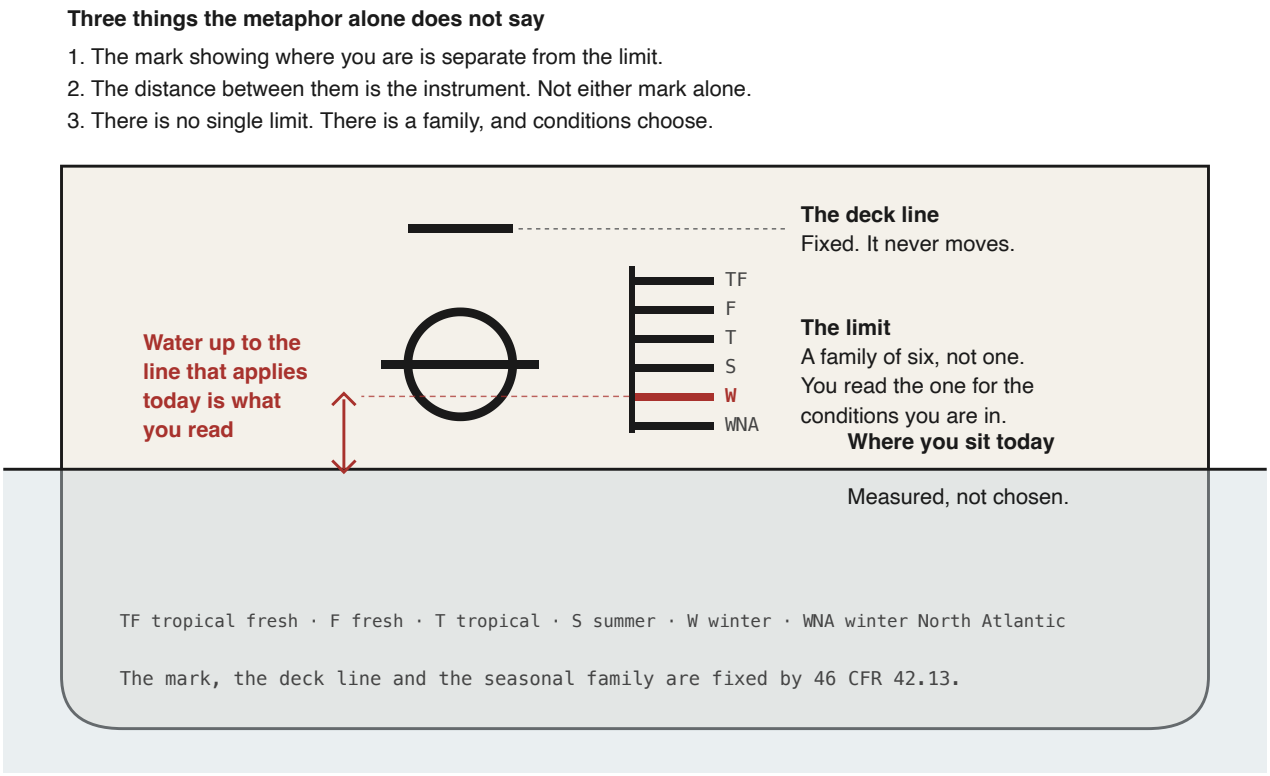


Figure 7: the deck line above, the family of six below it, the water below that. The gap from the water up to whichever line applies today is the reading, and it says how much more you can take.

The deck line is fixed. It never moves. Below it sits the load line mark, a ring with a horizontal line through it. One mark carries the safe maximum. The other is only a reference. Neither one tells you where you are. Both are fixed by regulation, down to their size and their position on the hull, so that anyone can read any ship the same way [38].

The second thing is the whole point. There are three elements, not two. The deck line is the fixed reference. The mark is the safe maximum. The waterline is where you sit today, measured and never chosen. The reading is the distance from the water up to the line that applies today, and it exists to tell you how much more you can take without losing quality or control.

Keep that separate from the other distance in the picture, because the two get confused and they do different jobs. Measure down from the upper edge of the deck line to the center of the ring and you have the assigned summer freeboard. That is a number computed in advance and it fixes where the mark goes. It never moves once the ship is certified, and it tells you nothing about today.

So the freeboard sets the limit. The gap from the water up to today's line is what you read. Read either mark on its own and you learn nothing. Read the gap and you know how loaded you are and how much more you can take.

The third thing is that the safe load changes with conditions. Six lines run off a vertical bar beside the ring. They are lettered for summer, winter, winter North Atlantic, tropical, fresh water and tropical fresh water. Gentler conditions allow a deeper load. You read the line for the conditions you are in.

The rule fixes the letters and the geometry. Nothing else. Where each line sits is a freeboard computed elsewhere. The limits are calculated. They are not drawn by preference.

The rule forbids departing for sea with the applicable line under water. Everything above that line is load the ship is built to carry, and the mark is what lets her carry it. Crossing is a defined state, not a judgment call. The exceptions are computed allowances, written into the certificate. They are not discretion in the moment.

Now the honest part. The limit in this paper is Drucker's, not mine. He argues for one task at a time. A minority who need a change of pace manage two. Almost nobody manages three. Those hedges stay.

No published figure covers this work, and publishing a universal number would be the wrong kind of help. Section 1.2 is the reason: capacity is personal, and two people watching the same work do not face the same bound. A number I gave you would be a number you borrowed, which is precisely what this section is telling you to stop doing.

Drucker's stays because of where it comes from. It is the most durable figure anyone has put on this, it has been argued with for sixty years, and it is worth having in front of you as a reference point. It is not your answer. Yours has to be calculated, from your own log, against the work you actually do.

So the mark maps onto the Method one part at a time.

- The deck line is the fixed reference. Here that is the work you own, and it never moves.
- The ring is the safe maximum. Here that is the number, and the number is not set.
- The gap from the water up to today's line is the reading. Here that is the distance between what you can take in and what is open, which is the room you have left.
- The lettered lines are conditions. Here they are your workload and your fatigue. They do not move the disc. They change which line you are measuring to, which is why the same open count can be fine in one week and too much in the next.
- Submergence is a defined state. On a ship it is the moment the line touches the water, and she may not sail. Here it is the moment your open work passes the number you read off your own log, and that number is the part you supply.

3.2 Where the instrument sits

You cannot set the mark from documents. You observe first, and three instruments do the looking.

A counter reads what you are holding. **A block log** reads where your time actually goes. **An intake log** reads what arrives, and when. The first two you can run this week. The third has to be built, and it is the one the mark most needs.

Three instruments read three different moments.

Run the instruments together before a start rule begins refusing work. Otherwise the rule grades its own homework.

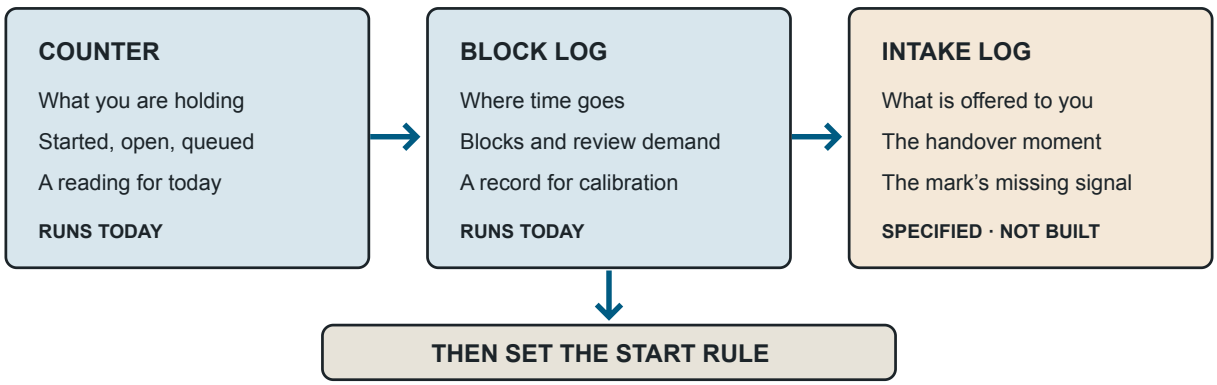


Figure 8: the three instruments observe different moments. Use them together before any start rule begins refusing work.

Two different controls are easy to confuse, so name them apart. This subsection is about the start rule, which decides when work begins. Section 2.2 is about acceptance, which decides what comes back. Different triggers, different moments, and both exist so you can be more effective without losing quality or control.

The first is the counter from 2.3, and it runs today. It queries the work you have already left behind: things started and not finished, decisions still open, items sitting in a queue.

Close to a to-do list, with one difference that matters. Nobody types into it. A to-do list holds what you remembered to write down, which is why it flatters you on the weeks you were busiest. The counter reads the traces the work leaves in the tools you already use, so it counts the things you forgot as well as the things you logged.

What it still cannot do is tell you what holding them costs, and it reports when something landed rather than when the thinking happened. So the counter is a report, and a report is not a look.

The second is a block log, and it is the one that reads time. Execution stopped costing you hours. Reviewing and deciding did not. A block is a stretch of work long enough to produce anything.

You write each one down as you take it, not at the end of the day and never from memory, because recalled hours are generous in one direction. Three to four weeks of that. By hand if you like, or dictated, or captured by an agent watching what you actually opened and for how long. The Method cares that the record is made as it happens. It does not care who or what holds the pen, and this is exactly the kind of clerical work you should be handing over.

Beside it, a short retrospective. Daily, then weekly. This is the part that makes the log worth keeping, and it is the piece people leave out.

A block log records what you did. It cannot tell you whether the week was any good, and that is exactly what you will need from it later. Four weeks on, a week that ran badly and a week that ran well look identical in a list of blocks.

So at the end of the day, three lines. What went well. What did not. What you would change. Two minutes, and most days it is dull, which is fine. The dull days are the baseline the bad

ones stand out against.

This is a good place to let an agent lead. Three questions asked at the end of the day, and your answers written down as you give them. You are far more likely to answer a question somebody asked you than to remember to ask it of yourself.

At the end of the week, the same three questions asked of the week rather than the day, and one more. Which days would you repeat, and which would you not.

That last question is the one doing the work. When you come to read the log you will be hunting for the weeks that worked, and by then you will not remember which those were. The retrospective is you telling your future self, while you still know.

Now the objection. A ritual that depends on you remembering to type something tends not to last. This one asks for something every day, so why would it survive?

Because there is almost nothing to it. It is three lines of plain text rather than fields in a form, so there is nothing to get wrong. It can be spoken. And once an agent is doing the asking, it turns from a thing you must remember into a thing that arrives. That is the shape worth building: the tool does the asking and the filing, and you do the answering.

If it still dies, that is a finding rather than a failure. A retrospective you cannot sustain for four weeks is telling you something about the week, and it is the same something the log would have told you.

What you do with all of it when the four weeks are up. This is the part that turns the exercise into a number, and it takes about an hour.

Lay the weeks side by side, with the retrospectives beside the blocks, and answer four questions.

How many blocks did a good week actually hold, counting only blocks long enough to finish something? How many of those went to reviewing and deciding rather than producing? What was open on the weeks you said you would repeat, and what was open on the ones you said you would not? And which week, if any, you would call a failure.

The retrospective is what makes the third question answerable. Without it you are guessing at which weeks were good, and you will guess in favor of the busy ones.

The number you want is the open count on your good weeks, not your average. The average includes the weeks you were coasting, and it is dragged the other way by the weeks you were under water. Neither is the week you want to plan around. Take the highest count that still sat inside a week you would repeat, and that is your first mark. It will be wrong. It is yours, and it is the first number in this paper that is.

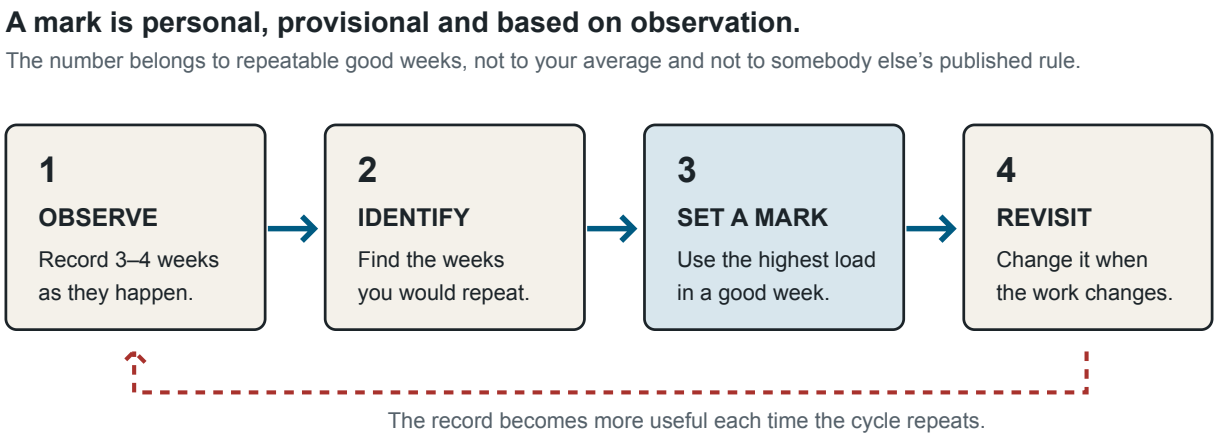


Figure 9: observe first. A provisional mark comes from weeks you would repeat, then changes as the work changes.

This works for whichever unit you chose, not just this one. Section 2.3 offers six, and the worked version above counts open decisions because that is the one with a reading today. Substitute your own without changing anything else. If your unit is returning review demand, lay the weeks against how much came back to you and when. If it is consolidated block-hours, against how many long stretches you actually got rather than how many you booked.

Then re-read it every quarter, because the number moves when your work does.

An agent is useful here too. Hand it the log and ask it to lay out the weeks against what was open, and to name the weeks that break the pattern. You are looking for a threshold, and finding a threshold in your own records is exactly the sort of reading you no longer have to do by hand.

One practice stays separate from all of this, on purpose. Look at the work yourself, with no report and no agent in between. Put it on the calendar and keep it.

Here is what it is for. Everything above is a report about the work. None of it is the work. A counter can be counting the wrong thing, a log can be recording what you meant to do

rather than what you did, and neither will tell you so, because an instrument that has drifted reads exactly like one that has not. Going and looking is the only check on the instruments themselves.

It is the same instinct as the rule that follows, and the two belong together.

The rule that already stands. What you notice outranks what the reading counts. If something feels wrong in the work, that beats a clean count, because a clean count is not a clean week.

Hold that rule alongside the finding in 1.3, that your reviewing wears down without giving you any sense that it is wearing down, and the two of them argue. A rule built on noticing inherits exactly that blindness. Both stay. Noticing catches what the count cannot see, and the count catches the weeks when noticing has gone quiet. Neither one is trusted alone, and that is the point of having two.

What the start rule does. Agent work begins when the person who will set intent, review and decide has room to do it. Where several people could take it, the rule reads the one who will rule on the work.

The obvious objection is why not start more. Starting is free. The agents are good, they are getting better, and nothing physical stops you launching six more things this afternoon.

Nothing stops you, and that is the whole problem. The constraint was never their capacity. It is yours, and it did not move when theirs did. Load past your line and the work does not stop arriving. It stops being reviewed. Section 1.2 says exactly what happens to it then, and it is worth being blunt about: past that line the work ships on the agent's judgment rather than on yours.

Most of it will probably be fine. That is not the point. You will not know which parts were fine, you had no view of the ones that were not, and you own all of it either way. The ship does not sink at the mark. She just stops being a ship anyone has assured.

So the reading is measuring the thing that actually binds. Not whether the work can be produced. Whether you can still carry the result.

Which is why the rule keys to you rather than to them. Capacity, workload, fatigue. That is the one measured result in section 1 running this way. Workload and fatigue each reduce

the number of mistakes a reviewer catches, and how much authority the machine held moderated neither [15].

There is a second reason, and it is the uncomfortable one. A rule keyed to agent quality would never fire anyway. Steady reliability is what erodes your catch rate, so agents behaving well are the condition that produces the problem rather than the condition that prevents it. A trigger waiting for them to slip is waiting on a signal the problem itself suppresses.

The third instrument, and why nobody has built it. Call it the intake log. The counter reads what you are already holding. The intake log reads what arrives, and when it arrived.

An intake log is missing from standard tooling for a simple reason. The moment it watches is not a moment anything currently records. Your tools log what you started and what you finished, because those are things that happen to the work. Being handed something for a decision does not change the work at all, so from a tool's point of view nothing happened. The one event the mark depends on is the one event nobody instruments.

Section 2.3 showed why that matters. The counter measures your working state rather than your inbox, which is close enough in a team where every piece has a named reviewer, but uninformative when you work alone. The intake log closes that gap by watching the handover instead of the residue.

It is a small thing to build. The whole specification is five properties and it is in the second appendix, so that anyone who wants one can make it rather than wait for somebody else to.

The order the pieces go in matters, and only the last needs a number. First the tracking tools—the counter and the intake log—go in where the start rule will eventually sit. They record everything and refuse nothing. Second, they run together for three or four weeks to establish your baseline number. Third, the threshold and the clearance go on top.

Here is what going the other way costs you. Say you turn the rule on first, at a guess, and set it to five open pieces of work. From then on, the record can only ever show five. Weeks where you would have taken eight look identical to weeks where you would have taken five, because the sixth, seventh and eighth were never allowed to arrive. Run that for a month and read the record, and it will tell you that five is exactly right. Of course it does. It was never allowed to say anything else.

The instrument that records has to be running before the rule that refuses, or the rule ends up grading its own homework. Measure first, then set the limit against what you measured, and expect the limit to be somewhere you would not have guessed.

Where this stands today. The counter runs. The block log you can start this afternoon, though nothing yet captures it for you. The intake log does not exist at all, so nothing yet watches the incoming work itself.

Be clear about which half is missing, because they are different things. The Method is the decisions: sort the work, set the authority, name the evidence, watch your own load. You can start every one of them this morning, and most of this paper is about them. What takes building is keeping them going when you are busy. That is the tooling, and the tooling is the part that is short.

So the start rule today is one you apply by judgment rather than a line you read off an instrument. The judgment is the same judgment either way. What the tooling buys is that you do not have to be sharp to exercise it, on the day you are least sharp, which is the day it matters.

3.3 The sentence that logs a trade

Let us say it is Tuesday and you are in the office.

You come out of a planning meeting with a new priority. Migrate the reporting stack by the end of the quarter, and everyone agreed it was the right call, and it was. Nobody in the room said what comes off your plate to make space for it, because that was not what the meeting was about. Your week was already full on Monday. It is still full, and now it has a migration in it.

Six weeks later the migration is late, and so are two things nobody mentioned in the meeting.

The rule is one sentence, and you write it in the room. Every goal names what it displaces. Not the theme it belongs to, not the reason it matters. The specific work that stops, or slows, or moves, so this one can happen.

The reporting migration displaces the monthly board pack automation until the end of the quarter.

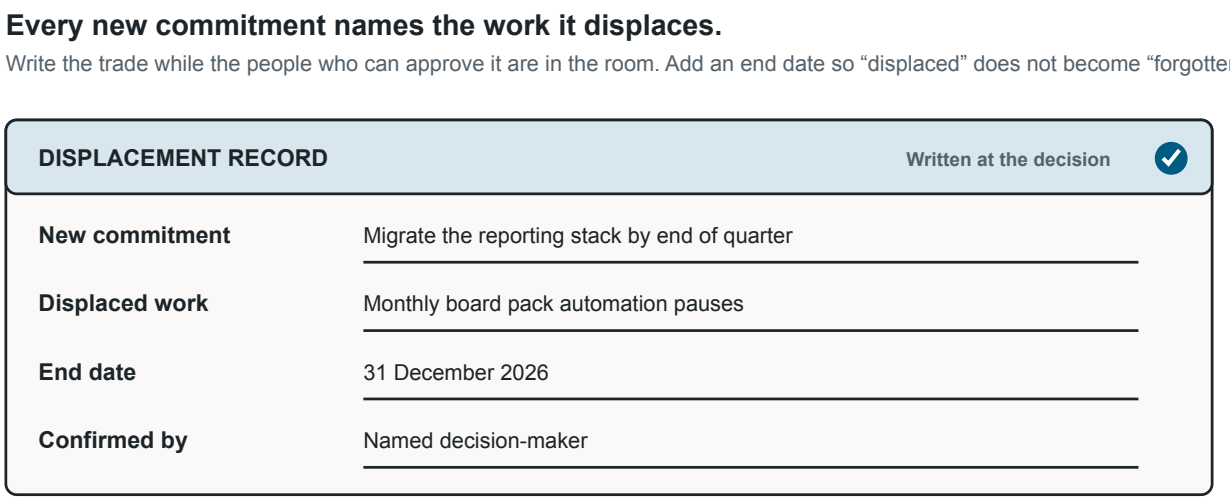


Figure 10: a useful displacement record names the new commitment, the work it moves, an end date, and the person who confirmed the trade.

That is the whole thing. It takes eight seconds and it changes the meeting, because the trade becomes visible while the people who can approve it are still in the room.

Why this rule and not a better one. Agents made producing cheap. They did not make your week longer. Every other control in this paper is about what you accept or how much authority you hand over, and none of them touches the one quantity that did not change. This is the only rule here that is about time, and it costs nothing to run.

It also catches the failure that is hardest to see from inside. Work does not usually get dropped by decision. It gets crowded out quietly, by things that each looked affordable on their own, and nobody notices until something is late. Approving one thing means stopping another whether or not anybody says so. The sentence just makes the organization say it.

An agent is genuinely useful here, and this is one of the places where it costs you nothing to ask. Give it your current commitments and the new goal, and ask what would have to give. It will name two or three things you were quietly planning to absorb, which is enough, because the point is to have the trade in front of you before you say yes rather than after.

Where it stops working. Two places, and both are honest limits rather than reasons to skip it.

Where your priorities are set above you, the sentence is a request rather than a decision. You still write it. It just travels upward as a question about what comes off, which is a better question than the one usually asked.

And it needs no tool, which is deliberate. The sentence works because it lives in a sentence, said out loud while the people who can approve the trade are still in the room. Nothing has to be entered anywhere for it to have done its job.

That matters, because a ritual that depends on you remembering to type something into a system tends not to last. I have watched two of mine go that way. A time column built to hold exactly this kind of number carries 149 rows and not one filled hour.

The failure there was the remembering, not the writing, and remembering is the part you can hand over. An agent reading the goal as it is set can ask the question at the moment it matters, propose what it thinks the trade is, and file the answer without you touching a field. That is the difference between a good intention and a process, and it is the smallest useful thing anyone could build around the Method.

Now the part that is not finished, because it matters.

A sentence logs a trade. It never refuses one. Nothing in it confirms that the displaced work actually stopped, and nothing in the Method yet emits the signal that would prove it.

So it can certify the failure it was written to catch. You write down that the migration displaces the board pack automation. The board pack automation never actually stops. Six weeks later everything is late, and the note now reads as evidence that you thought about it.

What turns it into a control is a number. With one, the claim is checkable: the displaced thing is off the list, and your open count came back under your limit. Without one, X displaces Y is unchecked prose. It reads like a control and it is a record. The number comes from the counter and the block log together, which is why 3.2 puts them first.

One thing you can add today, at no cost. Give the displacement an end date as well as a name. Work displaced for an unbounded stretch was never displaced. It was queued, and it will come back at a worse moment than the one you chose.

So write it anyway. It costs eight seconds, it puts the price of a yes in front of the person saying yes, and it is the only control in this paper that works in a room full of people who

have never read it. Read it as a record for now. It becomes a control the day there is a number behind it.

4. Conclusion

The question is how much you can load and still be effective, and how you would know.

Most of it you can act on today. Send away what a test can settle. Set the authority by cost and probability before the work. Name the evidence that will accept it, and accept on nothing else. Give each of the four stages its own amount of authority. None of that waits on an instrument.

Reading your own load is where the work is. Ashby gives the bound, and it is personal.

And none of it should end up being kept by hand. Every rule here is a decision, and decisions survive being made once. What does not survive is the upkeep, and the upkeep is what killed both of mine. So give the upkeep away. Let an agent hold the boundary sheet, watch the tripwire, ask the week's questions on a Friday, and tell you when the water has come up to the line. It keeps the apparatus and you keep the judgment, which is the only part of this that was ever yours alone.

Three things are worth knowing that we do not. How much one person can actually carry, which nobody has measured for this work. Whether findings from operators watching machines carry over to people reviewing agents, which is my analogy. And the one I keep returning to. When the agent is better at the task than you are, what does your review add?

Those are the edges. Naming them is how the next version gets better than this one.

Appendix: four pieces of work, run end to end

Every rule in this paper is easier to follow with a worked case beside it. These are four, chosen because they sit in different boxes of the matrix in 2.1.

A pricing table for a client proposal

Class. Costly to undo, and no test can settle it. Bottom right box.

Boundary. Inside. A quoted rate that reaches a client cannot be un-quoted, and the damage is commercial rather than clerical.

Authority before the work. The agent gathers and analyzes freely. It drafts the table. It does not send anything.

Evidence named before it goes out. Every rate traced to the rate card it came from, and the arithmetic reproduced from the inputs rather than restated.

Acceptance. You check the traces, then you send it yourself. That last step is not ceremony. The act nothing can verify is the act you perform.

Tripwire. A rate that differs from the card by any amount stops the work and comes to you, whatever the reason given.

Reconciling two ledgers at month end

Class. Cheap to undo, and a test settles it. Top left box.

Boundary. Outside. If the reconciliation is wrong it is wrong again next month, and nothing has left the building.

Authority. Full, through all four stages. The agent gathers, analyzes, decides and acts.

Evidence. The reconciliation balances, and the check can be rerun by anyone.

Acceptance. Accept the result without reading the work. Your reading adds nothing a balanced reconciliation has not already proved.

The one thing that stays yours. Whether these were the right two ledgers. No test can answer that, because pointing the test somewhere is what you did rather than what it checked.

A literature summary for a strategy paper

Class. Costly to undo, and a test settles part of it. Bottom left, mostly.

Boundary. Inside, because a fabricated citation that reaches a board survives every later correction.

Authority. Full on gathering and analysing. Nothing published without you.

Evidence. Every citation resolves, and every claim is checked against the source rather than against the summary of it. A second agent that did not write the summary does the checking, and returns the sentence it found.

Acceptance. On the resolved list. Not on how well it reads, which is the thing this class is designed to produce.

Tripwire. One unresolvable citation and the whole summary goes back, rather than that one line being cut.

A recurring internal status note

Class. Cheap to undo, and no test settles it. Top right box.

Boundary. Outside. Nobody is harmed by a clumsy status note, and next week brings another one.

Authority. Full up to sending. You still own what goes out under your name, and section 1.1 does not stop applying because the stakes are low.

Evidence. None named. Read it once and send it.

The trap in this box. This is the work most likely to eat your attention anyway, because it is easy to review and it feels like being responsible.

Appendix: what the intake log has to do

Section 3.2 says the intake log is the piece the mark most needs and the piece nobody has built. This is the whole specification, kept small on purpose so that anyone can build it rather than wait for somebody to.

- **Fire on the offer.** When a piece of work is put in front of a named person for a decision. Not when it is created, not when it is finished.
- **Record four things.** What it is, who it is for, when it arrived, and which class from the boundary sheet it belongs to.
- **Refuse nothing.** It is a sensor. A sensor that can refuse censors its own record, which is the grading-its-own-homework problem in 3.2.
- **Derive, never ask.** If it needs somebody to type something in, it will die, for the reason 3.3 gives.

- **Stay readable as a plain list.** So a person can count it, and an agent can lay it against a block log without anything having to be exported first.

That is the whole of it. Five properties, no schema worth arguing about, and nothing in it specific to any one tool or trade. It goes wherever your work already arrives.

Appendix: further reading

None of these is cited above. Each one goes further on one part of the paper. The note after each says what you get from it, and what to know about it before you lean on it.

Why review wears down. Section 1.

- Lisanne Bainbridge, "Ironies of Automation", **Automatica** 19, no. 6 (1983): 775-779, doi:10.1016/0005-1098(83)90046-8. Five pages arguing that watching a machine that is usually right is harder than doing the job, and that the skill you need when it fails fades while you watch. It argues rather than measures, and it is about control rooms and flight decks, so carrying it to agents is an analogy.
- David D. Woods, Sidney Dekker, Richard Cook, Leila Johannesen and Nadine Sarter, **Behind Human Error**, 2nd edition (Ashgate, 2010; CRC Press, 2017), doi:10.1201/9781315568935. The argument that what gets called human error is usually a symptom of how the system was built. Drawn from accident investigations rather than measurement.
- Daniel Kahneman, Olivier Sibony and Cass R. Sunstein, **Noise: A Flaw in Human Judgment** (Little, Brown Spark, 2021). How far careful people applying the same rules disagree, and how much further than their organizations expected. A popular book built on research, not a study itself.

Authority and delegation. Section 2.2.

- L. David Marquet, **Turn the Ship Around! A True Story of Turning Followers into Leaders** (Portfolio, 2013). A submarine captain's account of pushing decisions down to his crew. He hands over control as the crew's competence grows, where this paper sets authority by the risk of the work. A memoir, and Marquet now sells training built on it.
- Brad Anderson, Max Michaels and Rao Mikkilineni, "The Illusion of Permission: When AI Outruns Authority, Governance Must Be Native", Mindful AI Foundation, 2026, free from mindfulai.foundation. The case that an agent being able to reach a system is not permission to act on it, and that the check belongs inside the software. An advocacy

paper from a foundation two of its authors lead, promoting their own framework. Not peer reviewed.

Evidence and certification. Section 2.2.

- NASA/Navy Benchmarking Exchange, **Volume I: Interim Report, Navy Submarine Program Safety Assurance** (NASA and Naval Sea Systems Command, 20 December 2002), NASA Technical Reports Server 20030107501. How the submarine safety and naval reactors programs each assure safety, set beside NASA's approach. Free. Written by the organizations it describes, as an interim report.
- Columbia Accident Investigation Board, **Report, Volume I** (August 2003), NASA Technical Reports Server 20030066167. What happens when exceptions become routine, with the Navy's submarine program as the board's point of contrast. Free. The board's own findings.
- Diane Vaughan, **The Challenger Launch Decision: Risky Technology, Culture, and Deviance at NASA** (University of Chicago Press, 1996). How the line for acceptable risk moved one reasonable step at a time, until crossing it broke no rule. Worth reading beside the exception log in 2.2. One sociologist's study of one disaster.
- Nancy G. Leveson, **Engineering a Safer World: Systems Thinking Applied to Safety** (MIT Press, 2011; open access edition 2012, doi:10.7551/mitpress/8179.001.0001). Safety treated as a control problem across a whole system, with a chapter on SUBSAFE. It argues against scoring risk as cost times probability where software and people are involved, which 2.2 does, so read it as the objection to that part.

Load and flow. Sections 2.3 and 3.

- Donald G. Reinertsen, **The Principles of Product Development Flow** (Celeritas Publishing, 2009). The arithmetic of queues, and why too much work open at once slows everything behind it. The problem section 3 reads in one person, worked out for product teams. Self-published through his own company, and a practitioner's book rather than a study.
- Andrew S. Grove, **High Output Management** (Random House, 1983; Vintage, 1995). A working manual for managers by the man who ran Intel, including the idea that a manager's output is the output of the work they oversee. Useful now that anyone running agents is a manager. His own account of his own methods.

What this paper leaves out.

- Edward Honour, Joseph Lehman, Mohsin Ali and colleagues, "A Scalable Database Management System for Long-Term Institutional Memory, Human-AI Knowledge Sharing, and Contextual Recall", working paper (Zenodo, 2026), doi:10.5281/zenodo.20114119. This paper does not cover how agents keep what they learn. Honour's paper is where the case for a proper memory system is made. Read it knowing what it is: Honour maintains the system it describes, so it is a requirements case for his own product rather than neutral evidence.

About the author

Mathew Hager runs Running Fix, a consulting practice on how organizations get useful work out of AI agents. The Load Line Method is at loadlinemethod.com.

Sources

Numbers in the text point here. Where the paper names the author in the sentence, the number is placed at the first mention only. Anything marked as read second-hand says so, because the paper's own rule is that a source has to be checkable.

1. Raja Parasuraman and Victor Riley, "Humans and Automation: Use, Misuse, Disuse, Abuse", **Human Factors** 39, no. 2 (1997). The definition of automation quoted in the front matter. The 2000 paper adopts it from here, so this is the one to cite.
2. GitLab Inc., **GitLab Handbook**, public repository, "Communicating When Using Generative AI Tools". Captured 5 September 2026 at commit 4165803. The two rules used here are "Own your output" and its instruction not to use AI disclosure to deflect responsibility.
3. Hyman G. Rickover, testimony to the Joint Committee on Atomic Energy, 1961. The authenticated wording is that responsibility can reside only in a single individual, and that you may delegate it but it is still with you. Widely misquoted in longer forms.
4. Francis Duncan, **Rickover and the Nuclear Navy**, the official history commissioned by the Department of Energy, free in full from energy.gov. The source for headquarters running crew examinations, for the move to fleet examining boards after 1964, and for what Rickover kept when he stopped doing it himself.
5. W. Ross Ashby, **An Introduction to Cybernetics** (New York: John Wiley & Sons, 1956), section 11/7. The law of requisite variety. Cite by section rather than page: the page numbers in circulation come from a scan of the 1956 printing. It is a theorem, so it is evidence about nobody.

6. Ashby 1956, section 11/12. The channel form. A regulator can be no better as a regulator than it is as a channel of communication.
7. W. Ross Ashby, "Requisite variety and its implications for the control of complex systems", **Cybernetica** 1, no. 2 (1958). Ashby's own extension of the law to people, naming the manager. Also where he licenses approximate use, on the ground that only the existence of the limit is at stake.
8. Konstantinos Poulis and Efthimios Poulis, "Problematizing Fit and Survival: Transforming the Law of Requisite Variety Through Complexity Misalignment", **Academy of Management Review** 41, no. 3 (2016). The peer-reviewed objection to importing requisite variety into management. Read in abstract only.
9. Ashby 1956, section 12/5. A regulator acting only on errors it can already see cannot be perfect. This is the proof the paper leans on hardest.
10. Raja Parasuraman and Dietrich Manzey, "Complacency and Bias in Human Use of Automation: An Attentional Integration", **Human Factors** 52, no. 3 (2010). The field's standard review, and where I read the 1993 figures rather than in Parasuraman, Molloy and Singh's own paper.
11. Bagheri and Jamieson, 2004. Confirms the constant-versus-variable reliability effect. Reached through the 2010 review.
12. Singh, Sharma and Parasuraman, 2001. Sixty minutes of extra practice did not reduce the effect. Reached through the 2010 review.
13. Hans P. A. Van Dongen, Greg Maislin, Janet M. Mullington and David F. Dinges, "The cumulative cost of additional wakefulness", **Sleep** 26, no. 2 (2003). It measured reaction lapses, digit-symbol substitution and serial arithmetic. It did not measure judgment. A published erratum exists that I have not read.
14. Jason S. McCarley, Sarah P. Gyles, Kaitlyn R. Hankey, Fernanda Muñoz Gómez Andrade and Yusuke Yamani, "Deconstructing the vigilance decrement: Changes in bias, lapse rate, and guess rate, but not sensitivity", **Attention, Perception, & Psychophysics** (2026). Preregistered, not a registered report. It found no sensitivity loss, and the paper says so.
15. Griffiths and colleagues, 2025. 204 undergraduates, simulated air traffic control, thirty minute scenarios. Workload and fatigue each cut what a person caught; automation level moderated neither. Its authors call it a partial failure to replicate its predecessor.
16. Tzvetomir Blajev and William Curtis, **Final Report to Flight Safety Foundation: Go-Around Decision-Making and Execution Project** (Flight Safety Foundation, March 2017). The 3 percent compliance figure is derived from two other studies rather than measured, and the report says company compliance was never objectively verified. The manager survey was completed by 18.6 percent of site visitors.

17. Heleen van der Sijs, Jos Aarts, Arnold Vulto and Marc Berg, "Overriding of drug safety alerts in computerized physician order entry", **Journal of the American Medical Informatics Association** 13, no. 2 (2006). Carries both halves: the override rate, and the finding that overriding is mostly correct.
18. Brian L. Strom et al., "Unintended effects of a computerized physician order entry nearly hard-stop alert to prevent a drug interaction", **Archives of Internal Medicine** 170, no. 17 (2010). The trial that ended early after four patients had treatment delayed. Read in abstract only; the full text is paywalled.
19. Kristina Lång, Viktoria Josefsson, Anna-Maria Larsson et al., "Artificial intelligence-supported screen reading versus standard double reading in the Mammography Screening with Artificial Intelligence trial (MASAI)", **Lancet Oncology** 24, no. 8 (2023). An interim clinical safety analysis, not a final result.
20. Jamieson and Skraaning, 2020. One of two direct tests that failed to reproduce the cost of higher automation.
21. Bowden and colleagues, 2025. The second, which reversed the sign.
22. Linda Onnasch, Christopher D. Wickens, Huiyang Li and Dietrich Manzey, 2014. The meta-analysis is two results. Routine performance rises with automation level across sixteen studies. Performance on taking the work back falls across nine.
23. Merritt and colleagues, 2019. The complacency-proneness scale failed to replicate on 475 people.
24. Neville Moray and Toshiyuki Inagaki, 2000. Without a stated benchmark, complacency is a verdict rather than a measurement. The objection lands on anything the Method builds, which is why it is printed.
25. Kenneth J. Arrow and Anthony C. Fisher, "Environmental Preservation, Uncertainty, and Irreversibility", **Quarterly Journal of Economics** 88, no. 2 (1974): 312-319. Claude Henry proved the same result independently that year. Citation confirmed, wording not: the journal is paywalled and the readable scan carries no text layer, so this is paraphrased and never quoted.
26. Rear Admiral Paul E. Sullivan, testimony to the House Science Committee, 29 October 2003. The public source for the Thresher figures, for the certification boundary, for objective quality evidence, and for the separation of authorities. Read it knowing what it is: one man's account of his own program, given as advocacy two months after the Columbia board held it up as a model. No independent audit of the program is in the public record.
27. DORA, **Accelerate State of DevOps Report**, 2024 and 2025. Their factor is called instability. The 2025 report reverses the 2024 throughput finding, so only the instability

result is used here.

28. Raja Parasuraman, Thomas B. Sheridan and Christopher D. Wickens, "A Model for Types and Levels of Human Interaction with Automation", **IEEE Transactions on Systems, Man, and Cybernetics** 30, no. 3 (2000). The four stages, the risk line quoted in 2.2, and the error trapping recommendation. The authors state on page 294 that they defined no scale for the other three stages.
29. Tony Cox, "What's Wrong with Risk Matrices?", **Risk Analysis** 28, no. 2 (2008). Sixteen pages, free. It shows a matrix can rate a smaller risk above a larger one. It does not touch multiplying a real cost by a real probability.
30. Chief of Naval Operations, **OPNAVINST 3500.39D, Operational Risk Management**, 29 March 2018, cleared for public release. Severity in four categories and probability in five, combined into an alpha-numeric code rather than a product. The matrix is optional in its own words, the ranking is called a guide to relative priority rather than an order to follow, and no score in it triggers an automatic accept or reject. The five control options are in Enclosure 1. Read it knowing what it is not: it gates the accept decision on authority rather than on evidence, and authority is the half that does not transfer to one person who already holds all of it.
31. AXELOS, **ITIL 4 Change Enablement practice guide**, 9 January 2020. Public. Consensus practice sold by a training business, reporting no studies of its own, so it is cited for the shape of the rule and never as evidence.
32. Crocoll and Coury, 1990. Thirty-two people in four groups of eight, on an aircraft identification task. The design comes from the published abstract and the result through Parasuraman, because the full paper is paywalled. One study, eight per cell. Treat it as a hypothesis with a single test behind it.
33. Thomas B. Sheridan and William L. Verplank, **Human and Computer Control of Undersea Teleoperators** (1978). The ten-level scale, reprinted as Table I of the 2000 paper and read off that page. Written for decision and action selection only.
34. Aviation's minimum equipment list. Every deferred item is born with a date it expires, and the category carries the deadline rather than a conversation.
35. Eugene Gilbo, Scott Smith and Michael McKinney, "A New Approach to Monitoring and Alerting Congestion in Airspace Sectors", ATCA 59th Annual Conference Proceedings (2014). Demand is predicted per minute; the limit is assigned separately, one value per sector.
36. Peter F. Drucker, **The Effective Executive**, 50th anniversary edition. Cited by chapter, since this edition has no page numbers. Keep all three of his hedges: one task, two for a minority, almost nobody at three.

37. David Allen, **Getting Things Done**. Used for the structure only. Anything agreed and unresolved keeps presenting itself, including work done but never marked done.
38. 46 CFR Part 42, subpart 42.13 and section 42.07-10, implementing the International Convention on Load Lines 1966. The inch figures are the American ones. The Convention and the UK regulations are metric and were not fetched, so nothing here is converted.
39. Niels Johannes, "Plimsoll mark" (2025), photograph of an International load-line mark, CC BY-SA 4.0, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Plimsoll_mark.jpg.

Changes

- **Version 1.7, September 2026.** Reader review completion (Sharon's review, round 2). Sections 2 and 3 remove redundant internal documentation side trail, clarify trigger discipline around operator unease, replace compressed "offered" and "instruments" jargon with settled vocabulary, state tracking setup sequence cleanly, and remove personal build status admission in favor of the personal capacity principle.
- **Version 1.6, September 2026.** Reader review round (Sharon's review). Byline links to personal website. Summary cuts duplicate procedure, clarifies controls, and streamlines the displacement rule. Sections 1 and 2 clarify ambiguous pronoun referents, translate compressed source formulations (Ashby, Arrow & Fisher), state consequence of early trial stops, and cut redundant roadmap and reassurance lines.
- **Version 1.5, September 2026.** Two lines of work merged: a credited photograph of a real load-line mark now opens the paper, and eight new figures carry the decision aids.
- **Version 1.4, September 2026.** The working-version note under the byline is gone.
- **Version 1.3, September 2026.** Clarity edits from a reader. The three findings in section 1.3 now read as one series. "Erodes catching" becomes "erodes your catch rate" in all three places it appeared. Two more sentences take a colon. Spelling is American throughout, which it was not.
- **Version 1.2, September 2026.** First version published here. Two definitions are now set as quotations. Rickover's line now matches the hearing record.