

TEAM IQ

*why your best people
don't make your best teams*

Andrew Phillips

TeamIQ Ltd

Team IQ: why your best people don't make your best teams

Copyright © 2026 TeamIQ Ltd. All rights reserved.

First edition, 2026. Published by TeamIQ Ltd, registered in England and Wales. TEAM IQ Creator is a registered trademark, and is used in this book to refer to the method and the accompanying instruments.

No part of this publication may be reproduced, distributed or transmitted in any form without the prior written permission of the publisher, except for brief quotations in a review or in internal business documents, which are permitted with attribution.

ISBN (paperback): to be assigned

ISBN (ebook): to be assigned

Edition: 1.0

The exercises and instruments described in this book are published so that teams can use them without buying anything. You do not need to work with us to use them.

Every statistic in this book is either drawn from published research, named in the text and listed in the references, or identified as our own observation. Where a figure is quoted, it is quoted as published. Nothing here is a guarantee of a commercial result, and nothing here is professional advice for your particular circumstances.

Written in the United Kingdom in British English. Set in EB Garamond and Inter.

systemhelp.co.uk

Contents

- Why the focus is on teams 1
- Part One — Why capable teams underperform 8**
 - 1. The finding nobody acts on 9
- Part Two — The value question 18**
 - 2. Output is not value 19
- Part Three — The measurement cycle 28**
 - 3. The measurement cycle 29

How to use this book

This book has three jobs. The first is to convince you that the performance of a team is a measurable property of the team, not a sum of the people in it. The second is to give you six things to work on. The third is to give you a way to show the movement to whoever holds your budget.

Six parts follow. The first three set up the argument, the value question and the measurement cycle. Parts four, five and six are the six pillars in practice, and they are the working part of the book: each chapter ends with an exercise you can run with your team, something to measure, and one thing to do this week.

You can read it straight through in an evening or two. You will get more from it if you run the exercises as you go, on one real team, rather than making notes for later. Every exercise is designed for a single session of forty-five to sixty minutes, with the team in the room.

Nothing here requires you to buy a framework, a licence or a subscription. The instruments are published in the book and on the website, and they are yours to use.

A note on the evidence

Business writing has a credibility problem, and most of it comes from numbers that nobody can trace. Three rules were applied to every page of this book.

First, every published figure is named where it appears and listed in the references at the back. If you want to check a number, you can. Second, where a claim comes from our own work rather than from

published research, the text says so in the sentence, and the numbers come from programmes we can name. Third, where we could not source something, it is described rather than quantified.

That third rule cost us some good lines. A widely quoted figure about rework reduction appeared on our previous website without a source; we removed it rather than repeat it. You will find that attitude throughout: occasionally irritating, consistently honest, and the reason the arguments in this book survive being taken to somebody sceptical.

Why the focus is on teams

The transformation is complete. The framework is in place, the ceremonies are running, the tooling has been bought and the training has been delivered. Three years and a great deal of money later, delivery looks much as it did before: capable people, sensible plans, and a programme that still finishes late and costs more than the case predicted.

Ask why, and the answers are familiar. Priorities moved. A supplier slipped. The dependencies were harder than they looked. None of that is wrong, and none of it explains the thing leaders actually notice: that the same organisation can run one programme brilliantly and another badly, with much the same people and much the same process.

The difference is not in the process. It is in the team. This book is about that difference, why it is about to matter more than it has, and how to measure it and improve it.

Three pressures that are not going away

Speed. The interval between a decision and its consequence keeps shrinking. A competitor ships in weeks what used to take a year. A customer commitment made in a slide deck becomes a fact that the delivery team has to live with. When the market moves faster than

the plan, the plan stops being the constraint and the team's ability to decide well, quickly, becomes the constraint.

Complexity. Very little of the work now sits inside one team. A release crosses product, engineering, architecture, data, operations, security, legal, a supplier and a regulator. Each holds part of the truth. Nobody holds all of it. Complexity of that kind does not respond to effort; it responds to whether the people holding the parts can combine them into a decision.

Competition. Standing out now depends on making something better or different, not on doing the same thing more cheaply. That requires the people closest to the work to generate and test ideas — which is a property of the team, not of the individuals in it.

Add one more pressure, which is the one most organisations are actually feeling: being asked to do more, with less, faster.

The map is not the terrain

Process work has given what it can. That is not a criticism; it is an observation about what a process is. A process is a description of how work normally flows, captured at a point in time. It is a map. A map is useful, and a map is not the terrain. Walk a route from a map and you will meet the closed road, the washed-out bridge and the path that was never there.

Watch how fast the maps themselves move. The Scaled Agile Framework has been through six major versions since 2011, with many smaller releases in between. Every version is an improvement, and every version also means that whatever was rolled out, trained and embedded across a large organisation is now a version behind.

Rollout at that scale commonly takes a year or more. You can finish the training just as the map changes.

This is the trap: treating the framework as the answer, and the team as the thing that implements it. When the framework meets a situation it did not anticipate — which is most days — the framework cannot adapt. The team can. The German field marshal Helmuth von Moltke the Elder put it in a way that has survived: no plan of operations extends with certainty beyond the first encounter with the enemy's main strength. Mike Tyson later reduced it to a sentence everyone remembers.

Four kinds of problem, one habit of thinking

It helps to know what kind of problem you are in, because different kinds need different thinking.

- *Simple.* Manufacturing, largely. There is a best practice, and following it works. Instruction is appropriate.
- *Complicated.* Most software delivery. There are good practices and no single best one; expertise and analysis decide the route. Instruction alone stops working.
- *Complex.* The situation where nobody has a proven answer and the answer emerges from the people in it. Here you need everyone's context, not the most senior opinion.

- *Chaotic.* Something has broken and the priority is to stabilise. The habit that causes damage is treating every problem as simple: find the best practice, mandate it, and hold people to the instruction. That works in manufacturing and fails in software, because software

is always complex. A team that has only one way of thinking, whatever the situation, will be wrong in three of these four cases.

The alternative is not a better process. It is a team that can read the situation, change how it is thinking, and find a route that nobody has written down yet. Teams that can work without a map, and draw a new one.

What the research actually found

Two bodies of work sit behind this book. The first is from MIT: studies of *699 people working in groups of two to five*, published in *Science* in 2010. They found a general collective-intelligence factor — a property of the group, not of the individuals in it — that predicts how the group performs across a wide variety of tasks. It is *not strongly correlated with the average or maximum intelligence of the members*. It is correlated with members' social sensitivity, with how evenly conversational time is shared, and with the proportion of women in the group. The finding is uncomfortable for anyone whose answer to a struggling team is to add a cleverer person.

The second is Google's own study, begun in 2011 and running for about four years. It looked for what distinguishes effective teams and landed on five dynamics: psychological safety, dependability, structure and clarity, meaning, and impact. The strongest of the five was psychological safety — whether people can say what they actually think without paying a price for it.

Neither study gives you a method. MIT establishes that the property exists and can be measured. Google shows which conditions make it visible. What to do about it, week by week, is the gap this book

fills — and the six-pillar model in it is ours, not theirs. The research is the reason to take the subject seriously. The model is the reason you can act on it.

What we have seen for ourselves

Not every figure in this book comes from a study. In delivery work at a large UK financial services organisation, a programme we were part of saw rework fall by around 47% and the time taken to reach a decision fall by around 60%.

We include it because it is the closest thing we have to a before-and-after from our own practice, and we mark it for what it is: an observation from one organisation, on one programme, in work we were part of. It is not a benchmark, it is not a controlled measurement, and it is not a promise about your team. The organisation is not named, and there is no published result to point you at. Take it as a reason to run the measurement yourself, which the rest of this book shows you how to do. One thing should be said plainly, because it is usually left out. The collective-intelligence finding is contested. It has been replicated — a 2021 meta-analysis in *PNAS* covering 22 studies, 5,279 people in 1,356 groups found strong support for a general factor, and found that the group's collaboration process mattered more than the average skill of its members — and it has been challenged. A 2017 study in *Intelligence* found that individual IQ accounted for around 80% of the variance in group performance, and a 2024 study in *PLOS ONE* failed to replicate the three predictors.

What is settled is the operational point: some teams reliably out-perform what their members' averages would predict. What is still

argued about is the precise instrument. Our claim is the narrow one, and it does not depend on that argument being resolved: inspect the conditions, measure them, and see whether the number moves.

Why artificial intelligence makes this more urgent

Artificial intelligence is the most useful tool to arrive in delivery work for a generation. It is also, in the wrong place, a multiplier of the wrong thing.

An AI tool applied to a broken process does not repair the process. It produces the wrong output faster, in more volume, with more confidence, and it makes the mistake harder to spot because it arrives well presented. Applied to a team that cannot say what it actually knows, it accelerates the thing the team already does: generating material nobody has checked and decisions nobody agreed.

The tool is not the variable. The team is. A team with high collective intelligence will ask the model better questions, judge its output against evidence, notice what looks plausible and is not, and integrate it into a decision. A team without it will generate more, faster, and find out later. The same is true of every previous accelerator: cloud, automation, offshoring, a new framework. Each one increased the consequence of how well the team thought together.

What this book is

This book is not against process. It is against the assumption that process is the lever.

It sets out six pillars — Target, Expression, Accountability, Mindset, Insight and Quality — a five-stage measurement cycle, and fourteen instruments that turn each pillar into something a team can inspect

and change. Every chapter ends with an exercise you can run in one session, something to measure, and one thing to do this week. None of it requires a licence, a subscription or a purchase.

The claim is modest and testable: collective intelligence is a property of a team, it can be measured, and the conditions that produce it can be improved deliberately. If that is true, then the fastest performance gain available to most organisations is not a new framework. It is the team they already have.

Before you start

One thing, which takes ten minutes and no meeting.

Think of a decision your team got wrong in the last quarter — not a disaster, just a decision that had to be revisited, or that produced work nobody wanted. Write down two things: what information existed somewhere in the team at the time, and who held it. Then ask whether the decision could have been made correctly if everyone had pooled what they already knew.

Almost always, it could. That is the subject of this book.

Part One — Why capable teams underperform

1. The finding nobody acts on

At the start of the delivery meeting, the room looks strong. The product lead knows the customer case. The architect can explain the technical constraints. The delivery manager has the plan, risks and dependencies. The operations lead knows where the handover will hurt. Each person has a sensible update and has done the work expected of them.

Then the discussion moves to a decision. A dependency has changed, a supplier date has slipped, and the release still has a public commitment. The product lead protects scope. The architect protects the design. Operations asks for a longer proving period. The delivery manager asks for a revised date. Nobody is wrong. Nobody can show how the choices fit together.

The meeting ends with actions, each assigned to a capable person. The status report will be refreshed. A decision will be taken elsewhere. By the next meeting, the programme has moved, but not in a direction that anyone chose together. This is not a failure of effort or talent. It is good people producing a result that falls short of their combined judgement.

The principle

Collective intelligence is a measurable property of a team: its ability to perform across work that requires people to share information,

reason together, decide and adapt. It is not a compliment for a friendly team, nor a claim that every member thinks alike. It concerns what the group can reliably produce when the work needs more than one perspective.

The MIT collective intelligence studies found a group-level capability not accounted for simply by the intelligence of individual members. Google's Project Aristotle reached a related managerial conclusion: what happens between team members matters to effectiveness; selecting impressive individuals is not a sufficient design for a dependable team.

That finding is routinely acknowledged, then ignored in the machinery of an organisation. Recruitment inspects qualifications, experience and individual performance. Appraisal rewards the person who rescues difficult work. Promotion panels look for personal authority, subject knowledge and visible results. These are reasonable things to inspect. They are easier to inspect than the quality of a team conversation, the clarity of a handover, or the way a group changes course when its evidence changes.

The mistake comes in the inference. A leader sees capable people and assumes the team will make capable decisions. A programme is staffed with specialists and assumed to be safe. Work is divided, names are placed beside tasks, and the result is treated as the sum of the people. It does not follow. A group can contain expertise and still fail to bring it into the decision. It can be busy and leave a consequential assumption untested. It can have an owner for every task and nobody accountable for the outcome across those tasks.

The cost is disguised as ordinary delivery friction: late discovery, rework, slow escalation, decisions revisited after work starts, and senior people drawn back into matters the team ought to resolve. A weekly status report may show activity. It cannot tell you whether people share a target, have heard the same evidence, or will act together when a constraint shifts.

The six pillars in this book are *Target*, *Expression*, *Accountability*, *Mindset*, *Insight* and *Quality*. They make a team's intended result, communication, ownership, learning, information and standard of value visible. The rest of the book takes each pillar in turn, turns it into habits and artefacts, and uses the TEAM IQ Measurement Cycle to choose where to improve next.

The six-pillar model and its practical measures are ours, not a published result from MIT or Google. Our claim is modest: inspect and improve the conditions through which a team sets a target, exchanges evidence, owns outcomes, learns, makes decisions and judges quality. You then have a disciplined way to raise collective capability and see whether it is happening.

What it looks like when it is missing

Return to the delivery meeting. The product lead has a customer commitment in a slide deck, but the team has not seen the evidence behind it. The architect has logged a design concern in a document read only by the technical group. Operations has a service risk in a separate register. The delivery manager has a plan whose dates assume all three views can be satisfied.

When the supplier date changes, people speak from their own artefact. The discussion becomes a negotiation between local commitments. The architect protects a design decision because its effect on the customer target is unclear. The product lead protects scope because the service risk is not visible in prioritisation. Operations is asked for an answer after the decision has hardened.

The actions are individually tidy. Someone will ask the supplier for a revised date. Someone will estimate an alternative. Someone will update the plan. What is absent is a shared outcome, the evidence that should alter the decision, the person who can make the call, and the point at which the team will review the consequence. The group has distributed work without distributing understanding.

This can look orderly from outside. Meetings occur, work is assigned and papers are produced. It becomes expensive because uncertainty travels quietly between roles until it arrives as a late defect, a blocked release or a request for an executive decision.

What it looks like when it works

In the same meeting, the changed supplier date is placed beside a visible target that names the customer or business change the release must create. The team has agreed the evidence that would justify a change in scope, date or design. The product lead explains the value at risk. The architect explains the constraint in terms the whole room can use. Operations identifies the condition required before the service can change.

The discussion is still difficult. There may be a trade-off between date, scope and assurance. The team is working on one decision

rather than defending private versions of it. A concern is information that may alter the route to the shared outcome, not a challenge to someone's authority.

The meeting ends with a decision record: what has been decided, the evidence used, the open assumption, the owner of the next test and when the group will read the result. The delivery manager can change the plan because the reason is clear. The product lead can explain it to a sponsor without hiding uncertainty. Operations is involved before the handover.

This is not consensus theatre. A team can disagree sharply and still be collectively intelligent if it makes its reasoning visible, uses relevant expertise and commits to clear conditions for revisiting a course. The aim is a team that can think and act together when the original plan no longer fits the work.

How to do it

Do not begin by rating personalities. Begin with live work that exposed a gap between individual effort and team performance: a delayed decision, failed handover, late issue or differently interpreted priority. Inspect the way the team worked, not a person to blame.

1. *Choose one outcome, not a general complaint.* Put a live delivery outcome in the centre. State the customer, business, risk or service change required; the date or decision point; and the consequence if it is not achieved. A label such as poor communication gives the group nothing observable to inspect.
2. *Collect the existing evidence before the meeting.* Bring the plan, decision record, risk entry, customer information, service data and work board. Do not produce a new pack to make the team look

- aligned. Gaps between the existing artefacts are evidence. If a key statement exists only in one person's head, record it as a finding.
3. *Ask for individual accounts before group discussion.* Give each person time to write, without consulting colleagues: the outcome they believe the team is pursuing; evidence they trust; the main uncertainty; the decision needed; and who owns the next move. This exposes differences before the most senior or confident voice sets the answer.
 4. *Compare the accounts against the work.* Put answers beside the artefacts. Look for differences that would change a decision or fail a handover. Is the target the same in each account? Is a risk known to people who must alter the plan? Does the named owner have authority, information and support? Does a technical or operational condition appear only after a business commitment? Record the difference as a condition in the system, not a flaw in a person.
 5. *Make the next decision explicit.* For the most material difference, agree the decision, evidence still needed, person obtaining it, contributors and review date. Write the assumption plainly. A decision with no stated assumption cannot be intelligently revised; it can only be defended or abandoned.
 6. *Use the TEAM IQ Measurement Cycle to keep evidence connected.* The cycle is not a reporting layer added after delivery. It makes intentions, evidence, ownership, learning, decisions and outcomes visible in the work. At the end of an interval, read outcome, evidence, decision and learning together. Then decide whether the constraint concerns Target, Expression, Accountabil-

ity, Mindset, Insight or Quality. Work on that before adding another ceremony.

7. *Read patterns, not isolated incidents.* One poor meeting may be a timing problem. A repeated difference between the plan's target, a risk register's evidence and a decision's assumption is a team pattern. Look across live decisions before changing a role, meeting or process. Ask whether the team can make the connection without a senior person stitching the pieces together.
8. *Keep the intervention small and observable.* Agree one change to test in the next interval: a shared target in a decision record, an operations review before commitment, or a named evidence owner. Decide what you will look for. If it does not improve the quality or timing of the decision, keep the learning and change the practice.

This procedure is our applied method, built around the course's Measurement Cycle. It is not a clinical assessment and does not turn a team into a score. It exposes whether the team can connect the information and ownership required for the work. That is a better starting point than a general request to collaborate more.

The exercise

Run a shared-outcome review with the intact team around one live decision or delivery outcome. This is a practical book exercise, not a validated research instrument. Prepare a wall or shared board with space for target, evidence, uncertainty, decision, ownership, learning and outcome. Bring the plan, risk material, decision record

and customer or operational evidence. Ask only people who affect the decision to attend.

Allow five minutes to state the outcome and decision point. Give each participant ten minutes to write their account of the target, evidence, uncertainty and expected owner. Keep this silent. Find differences before discussion smooths them away.

Spend fifteen minutes placing accounts and artefacts on the board. Group similar statements, but do not force agreement. Mark differences that could change the route, timing, scope, risk or standard of the outcome. Use fifteen minutes to select the most material difference. Agree the evidence to settle or narrow it, who will obtain it, contributors, and when the team will return to the decision.

Use the final ten minutes to capture a one-page review record: shared outcome, unresolved assumption, next evidence action, named ownership, review point and the likely constraining pillar. Read it aloud. If anyone cannot explain the outcome or their next move, correct it before leaving.

Save a photograph or export of the board with the record for the next review.

The session takes fifty-five minutes. Its output is not a verdict on team quality. It shows where understanding breaks and tests whether a clearer connection improves the next decision.

What to measure

The first artefact is a one-page review record for a live outcome, used in the TEAM IQ Measurement Cycle. It links the intended result with available evidence, open uncertainty, decision, owner, learning and observed outcome. Keep the original record. The comparison

between what the team assumed and what happened makes learning visible.

Read it at the next delivery review and whenever the decision changes. Look for recurring breaks: differently explained targets, evidence after commitment, ownership without authority, or learning that never changes a decision. These signal a pillar for attention, not individual worth.

This is a TEAM IQ design choice, not a published universal score. At this stage, measure whether the team can show its reasoning, test it and alter work when evidence requires it.

One thing to do this week

Take a delayed or revisited decision. Ask each person involved to write the outcome, evidence they trust and uncertainty that matters. Compare the answers before anyone explains them. Do not repair every difference. Choose the one most likely to alter the decision, name the evidence needed and set the next review point. You will have a clearer diagnosis than a status update.

Part Two — The value question

2. Output is not value

The monthly delivery review starts well. The team has a full status pack, a release calendar, a burn-up chart and a page of completed work. The delivery lead has highlighted the items that went live, the tickets that were closed and the documents that passed review. The room can see that the team has been busy and that work has moved through its process.

The person funding the work listens, then asks their portfolio lead to say what is different outside the team as a result. There is a pause. Someone points to the release notes. Someone else explains that a dependency was removed and that the backlog is healthier. Those things may be useful, but they do not tell the funder whether a customer can now do something they could not do before, whether a business cost has fallen, or whether a material risk has changed.

The meeting turns into a defence of effort. The team has done what it said it would do. The funder has not received an account of what their money changed. Both statements can be true, which is why this conversation so often ends with more reporting rather than a better decision.

The principle

Output is the work a team produces: a release, a closed ticket, a set of story points, a design, a policy, a test result or a document. It

is visible because the team controls most of it. Output matters. You need to know whether work is moving and whether technical standards are being met. It is not, on its own, a reason to continue funding a team.

Value is a meaningful change for someone who cares about the result. That person may be a customer, a colleague handling an operational task, a regulator, a business owner or the organisation that carries the cost and risk. Value is not a feature list with a better label. It is the difference the feature, service or decision makes when it is used.

A senior reader will recognise the distinction. A team may release a self-service form. The output is the form and the integration behind it. The value may be that a customer completes a request without calling the contact centre, that the contact-centre colleague has fewer avoidable contacts, or that a request is completed more accurately. A team may also close a large batch of security findings. The output is the closed findings. The value is lower exposure to an identified risk, provided the closure is real and the remaining risk is understood.

The value question is plain: for whom will something change, what will change and how will you know? Add timing and cost to create and sustain it. Until a team can answer, it cannot be measured in a way that settles a performance conversation. Every available measure will describe activity: items started, items completed, time booked, tests run or pages produced. Those measures can be accurate and still leave the main question unanswered.

That is why arguments about performance become unwinnable. One leader points to delivery volume; another to an unchanged busi-

ness result; finance to cost. Each has a partial view, but none shares a defined result against which the team can be judged. The argument is then conducted through confidence, seniority and slide quality.

Google's team-effectiveness research cautions that measures such as lines of code and bugs fixed can mislead when used alone. The point is not to discard operational measures. It is to put them in their proper place: as evidence and guardrails around the result that matters. A team needs enough evidence to show that it delivered the right value, to the right standard, at the right cost and at the right time.

What it looks like when it is missing

In the next review, the same team returns with a fuller dashboard. The programme office has added separate counts for releases, defects, test cases, approvals and work in progress. The delivery lead can show that the queue is moving faster, but no one has agreed which customer or business result that movement should serve.

Work is selected because it is ready, because a senior stakeholder asked for it, or because it will make the plan look healthier. A feature reaches production and is treated as complete at deployment. Support receives new questions, operations changes a manual work-around and customers find another route, but those signals sit in different meetings. The team has no reason to bring them together because the work item is already closed.

Cost is discussed only when a request needs approval. The build estimate is compared with the remaining budget, while maintenance, training, support, security and retirement are treated as someone else's concern. When the outcome is weak, the response is to ask

for more features, faster delivery or a more detailed plan. No one can show whether that extra activity is likely to improve the result. No one owns the connection between delivery data and the result the work was meant to change.

What it looks like when it works

The team enters the review with a short statement at the top of its pack. It names the people affected, the change they should experience, the evidence that will show it and the date by which the team expects to learn. Its delivery measures are still there, but they are no longer the headline.

For an internal service, the team might state that people making a particular request should complete it without rekeying information, and that the organisation will see fewer avoidable contacts and less manual handling. The team brings task-completion evidence, operational data and a technical guardrail. If completion improves but service demand does not, the team investigates rather than declaring victory. If a release causes a reliability issue, the guardrail makes that visible even if the customer measure initially improves.

The funder can now ask useful questions. Is this the change we still need? Is the evidence credible? What will it take to achieve and sustain it? What should stop if this becomes the priority? The conversation moves from whether the team worked hard to whether continuing the work is a sound use of capacity. The statement stays open until the relevant evidence has been read and acted on.

How to do it

Use this method for a live workstream where its effects can be observed. It is a practical TEAM IQ method, not a claim that a sentence creates value. Its purpose is to give the team and funder a shared result to test.

1. *Choose one workstream.* Select a live stream, not the whole portfolio or a single technical task. Bring its objective, operating data and the person who can explain its funding. One stream prevents a vague statement that tries to cover everything.
2. *Name the person who must experience the change.* Write down the customer, operational colleague, business owner or other affected group. Describe their present situation in observable terms. Do not begin with the feature. Begin with the friction, cost, delay, risk or missed opportunity the work is intended to change. If the team cannot name a group and a current condition, pause. It has a delivery request, not yet a value case.
3. *Write the change and the evidence.* Use this form: For [named group], change [current condition] to [desired condition] by [timeframe], shown by [evidence], while maintaining [necessary guardrail]. Make every field specific enough that another leader could disagree with it. A statement that says an experience will be better is not testable. A statement that says a named group can complete a named task with fewer avoidable contacts, shown through service data and task evidence, can be tested.
4. *Set the threshold and the cost view.* Apply the 7 C's Value Engineering Standard (CVE) before development begins. Define what good enough means for the customer result and the relevant quality requirements, such as accessibility, reliability, accuracy,

security, performance or maintainability. Record the guardrails that cannot be traded away. Then make the unit economics visible. A delivery team costs more than build effort: include people, contractors where used, platforms, licences, supplier commitments, support, operations, security, training, maintenance and retirement. Bring costs held elsewhere into the conversation.

The board-level view is value per team. Set the value achieved in the period against the full cost of keeping that team available for the period, and identify costs that the change creates, removes or avoids over its life. Do not turn uncertain effects into a precise financial claim. State the assumptions, distinguish evidence from expectation and revise the view as use and operating data arrive. A cheap build that creates recurring support work may be poor value; a larger initial outlay may be justified where it removes recurring cost or material harm.

1. *Test the statement before committing more capacity.* Ask a leader who did not write it to explain the promised change. Ask the funding owner whether the change would alter a priority decision. Ask people closest to delivery whether evidence can be collected without a new reporting programme. Finally, ask what result would show that the work should be stopped, changed or not extended. If the team cannot answer, rewrite the statement. Make uncertainty part of the decision.
2. *Review results, not just release status.* Build a 7 C's Outcome Metrics (COM) scorecard around the statement: at least one customer measure, one business measure and one technical guardrail. Review the evidence after release and at a cadence that suits the

decision. Use it to scale, improve, reposition or retire the work. Retain delivery measures, but read them as leading evidence, not as a declaration of value.

The exercise

Run this as a fifty-five-minute session with the manager, delivery team and, where possible, funding owner. Prepare one active workstream, its objective and any customer, service, finance or operational evidence. Do not prepare a polished business case. Expose what is known and assumed.

1. *Set the workstream and affected group — 10 minutes.* The manager names the stream and group expected to benefit. Members write the current condition independently: what happens now, where the friction is and who bears the consequence. Compare descriptions and agree one that can be observed.
2. *Draft the value statement — 15 minutes.* Use the form in the method: group, current condition, desired condition, timeframe, evidence and guardrail. Read it aloud. Remove feature names unless needed to describe the change. Capture unresolved assumptions beside the draft.
3. *Check cost and quality — 15 minutes.* Use the 7 C's Value Engineering Standard (CVE) to identify the good-enough threshold and guardrails. List lifecycle costs that could change: build, support, operations, security, training, maintenance and retirement. Mark what is known, what needs validation and who owns it.
4. *Run the board test — 10 minutes.* One person who did not write the statement explains the value, the evidence and the cost trade-off without looking at the draft. The group then names the result that

would cause it to stop, change or lose priority. If the explanation is unclear, revise the statement once.

5. *Close with ownership — 5 minutes.* Capture the statement, evidence sources, guardrails, lifecycle-cost assumptions and next review date. Assign one owner for evidence collection and one for the funding conversation. The manager sends the record before the next delivery review.

The output is not a promise to produce a particular feature. It is a testable account of why the work deserves capacity and what the team will learn.

What to measure

The primary artefact is a value statement supported by a 7 C's Outcome Metrics (COM) scorecard. The statement records the affected group, the intended change, the evidence, the timeframe, the guardrail and the lifecycle-cost assumptions. The scorecard holds at least one customer measure, one business measure and one technical guardrail, so the team does not replace one narrow output count with another.

Use the 7 C's Value Engineering Standard (CVE) to record the agreed good-enough threshold before work begins. Where the work is substantial, add a Lifecycle Value Review to bring support, operations, finance, security and product perspectives into the cost view. Read the statement and scorecard in the normal delivery review, then read the outcome evidence at the decision cadence set by the workstream. The funding owner should see the same record, not a separate account of progress.

One thing to do this week

Choose the workstream that consumes the most attention in your next delivery review. Before the meeting, ask its funding owner to complete one sentence: the group whose situation should change, the change they expect and the evidence they would accept. Put that sentence at the top of the status pack. If nobody can write it clearly, make defining it the first item of work rather than adding another output chart.

Part Three — The measurement cycle

3. The measurement cycle

At 08:30 on a Tuesday, the portfolio office has the numbers ready. There is a delivery dashboard, a risk register, a burn-up chart, a financial forecast and a weekly pack for the executive meeting. Each team has sent its update. Green, amber and red boxes fill the screen.

A director asks a simple question: is the claims team getting better? The room goes quiet. The delivery lead can say how many items moved across the board last week. Finance can say what has been spent. The test lead can say how many defects were found. Nobody can show the change the team set out to create, the starting point, what the latest work was expected to alter, or what they will do because the evidence has changed.

The team is not short of measurement. It is short of a route from measurement to action. The reports describe activity in several places, with different owners and review dates. A problem noticed in a meeting is copied into the next pack, then the next. Work continues because it was planned, not because the latest evidence supports it. By Friday, another report is being prepared. The organisation has measured a great deal and still cannot say whether the team is improving.

The principle

The *TEAM IQ Measurement Cycle* turns an intended outcome into evidence and then the next decision. Its stages are Set, Communicate, Own, Learn, and Decide and improve. A decision at the end changes the target, work or risk view for the next pass.

Set establishes the valuable change, scale, baseline, target, guardrail, owner and review date. The team and sponsor agree it; the named metric owner maintains it. A shared target statement and T.A.R.G.E.T Value Index (TVI) are the artefacts. Skip Set and later measures may answer different questions.

Communicate makes claims, assumptions, evidence and changes visible. People doing the work bring evidence; the team maintains a common record. The C.L.E.A.R.L.Y Evidence Standard (CES) evidence log and Expression of Value Delta (EVD) update result. Skipped, a repeated opinion can pass as fact.

Own turns a hoped-for result into a commitment with accountability for progress and evidence. The primary owner is responsible; contributors make dependencies explicit. A D.R.E.A.M Outcome Contract (DOC) and, where authority or handoffs need clarity, Quantified D.R.E.A.M Boundaries (QDB), are the artefacts. Skip it and a target belongs to everyone in theory and no one when a trade-off is needed.

Learn tests assumptions that could affect the outcome or invalidate the plan. The outcome owner runs the test with the needed people; the team reviews it. A M.I.N.D.S.E.T Value Engineering (MVE) hypothesis and Risk-First Learning Loops (RLL) record leave the stage, alongside an updated risk view. Skip Learn and commitment rises as uncertainty does.

Decide and improve uses evidence to continue, change, stop or scale work. The named decision owner convenes the decision within the agreed boundary, with the team supplying relevant evidence. A D.I.R.E.C.T Value Routing (DVR) decision, IRT threshold, decision-log entry or DDA is the artefact. Skip this stage and the cycle produces reports that nobody uses; capacity and priorities remain unchanged.

The fourteen TEAM IQ measurement tools are instruments in the loop, not a separate reporting layer. Set uses T.A.R.G.E.T Value Index (TVI), Value-to-Effort Ratio (VER) and Target Risk Burn-Down (TRB). Communicate uses C.L.E.A.R.L.Y Evidence Standard (CES) and Expression of Value Delta (EVD); Own uses D.R.E.A.M Outcome Contract (DOC) and Quantified D.R.E.A.M Boundaries (QDB); Learn uses M.I.N.D.S.E.T Value Engineering (MVE) and Risk-First Learning Loops (RLL). Decide and improve uses D.I.R.E.C.T Value Routing (DVR), Insight Risk Triggers (IRT), D.I.R.E.C.T Decision Audits (DDA), 7 C's Value Engineering Standard (CVE) and 7 C's Outcome Metrics (COM). Each leaves a visible artefact for the next stage, rather than a second report beside the work.

Evidence moves deliberately. The TVI supplies the measure and guardrail for a DOC. CES separates knowledge from what needs testing. MVE and RLL update TRB, EVD and confidence. DVR routes capacity using value, effort, confidence and risk. The decision becomes a changed backlog, test, boundary or target, not a comment in a meeting record.

What it looks like when it is missing

The team starts the month with a list of commitments inherited from a planning meeting. The list has dates and named task owners, but no shared measure of the customer or business change expected. A dashboard shows completed work, defects and spend. When a senior stakeholder asks whether the work is helping, the team points to the growing list of completed items.

In the weekly meeting, people report that an integration is difficult. Nobody writes the assumption behind the design or decides what evidence would settle it. The risk stays amber because no one has defined a smallest test or named the decision it should inform. The work owner asks three colleagues for approval because the boundary of their authority is unclear.

At month end, the team has a slide describing the difficulty and a request for more time. It has no before-and-after measure, no record of what was learned and no explicit choice to stop, reduce or change the work. The next month begins with the same list, a revised date and another request for an update. Reporting has become the work that proves reporting is happening.

What it looks like when it works

The same team begins with one TVI: a stated outcome, current baseline, target date and guardrail. It names the person who maintains the measure and the person who owns each material outcome. A short DOC says what evidence will demonstrate progress, what support is needed and what the team will do if the expected movement does not occur.

In the weekly review, the owner uses EVD rather than an activity update. The update shows the before state, current state, learning and next action. One result has not moved. The team records the assumption in CES, puts a small test into the current work and updates the TRB. The difficult integration is now an exposure with a reduction experiment, not a paragraph in a status pack.

At the monthly decision review, the decision owner compares options through DVR. The evidence supports reducing one piece of work, testing another and continuing a third. Those choices are recorded with their rationale and review point. A trigger states what result would cause a pivot. The quarterly review looks across the decisions: whether targets still matter, whether the measures conflict, and whether the team is becoming more able to act on what it learns. The dashboard is shorter because each measure has a decision use.

How to do it

Run the first pass with one intact team and one material outcome. Do not build a portfolio dashboard or apply every tool. Establish a repeatable loop around work already under way.

1. *Days 1–3: Set the target and initial risk view.* Bring the sponsor, team lead, people closest to customer or service data, and the person who maintains the measure together. Write one TVI with value, scale, baseline, target, guardrail, owner and review date. List work that could move it. Use VER only for items on the same value scale. Create a TRB for the risks most likely to prevent the target, with a smallest risk-reduction experiment. Leave with the target statement, value-ranked work and initial TRB.

2. *Days 3–5: Make evidence and ownership testable.* Start a CES log for claims behind the selected work: supporting evidence, assumption, test, owner and date. Convert the next material commitment into a DOC: outcome, relevant measure, baseline and intended movement, evidence source, timeframe, dependency and learning rule. Use QDB where the owner needs a decision boundary or measurable handoff. Keep these live, not filed.
3. *Weeks 1–3: Run evidence updates and tests.* In the existing weekly meeting, reserve fifteen minutes for EVD. The work owner states what changed, which measure should move, the current observation, learning and next action. Run the MVE test or RLL experiment agreed in week one, favouring the risk with highest exposure. Update CES and TRB as results arrive. If no measure has moved, record learning and the next test rather than inventing progress.
4. *Week 4: route evidence into a decision.* Hold a monthly review with the decision owner and team. Read the TVI, EVD updates, TRB, CES assumptions and MVE or RLL results. Use DVR to compare next moves on value, effort, confidence and risk. Set IRT thresholds for a pause, pivot, escalation or further test. Record the choice, owner, rationale and review date: continue, adapt, stop or scale.
5. *At the quarter boundary: inspect the system, not only the result.* Use DDA on one material decision. Check purpose, evidence, assumptions, risks, options, authority and review point. Where quality is a trade-off, use CVE to set the value target, scale, good-enough threshold, guardrail, cost and stop-or-continue rule. Read

COM beside TVI for customer and business impact. Confirm or revise the next target.

Weekly is the rhythm for EVD, CES changes, active DOCs and TRB movement. Monthly is for a DVR decision, IRT review and capacity change. Quarterly is for DDA, target relevance and COM measures. This is a practical starting cadence, not a universal prescription. A fast operational team may need shorter decisions; a slow-moving outcome may not yield useful weekly movement. Keep the review date in the artefact and change cadence deliberately.

The exercise

Run a 60-minute cycle launch with one team before the first thirty-day pass. Prepare one live outcome that matters now, the best available baseline, the current work list, known risks, and one person able to confirm how the measure is sourced. Bring the sponsor or decision owner if a decision boundary will be needed. Use a shared screen or board that everyone can see.

1. *Set — 15 minutes.* Write the TVI together: value, scale, baseline, target, guardrail, owner and review date. Ask each person to explain the target in their own words. Capture the shared target statement and the one or two risks that most threaten it.
2. *Communicate — 10 minutes.* Choose the claim behind the next item of work. Create one CES entry: claim, available evidence, assumption, experiment, owner and learning needed. Capture the evidence source and date rather than a vague confidence statement.

3. *Own — 10 minutes.* Turn that item into a DOC. Name the outcome, measure, evidence source, timeframe, dependencies and learning rule. Confirm the primary owner and, if needed, the QDB decision boundary. Capture unresolved support needs separately.
4. *Learn — 10 minutes.* Design the smallest MVE or RLL test that could reduce the most consequential uncertainty. State what observation would change the team's view. Add it to the current work and capture the planned test date.
5. *Decide and improve — 10 minutes.* Agree the first decision point, decision owner and IRT threshold. Write the future DVR question: continue, change, stop or scale what? Capture where the EVD, TRB and decision log will be read.
6. *Close — 5 minutes.* Schedule the weekly evidence review and monthly decision review. Photograph or save the board. The launch does not replace the test; it makes the first test and the decision it should inform visible.

What to measure

The first measure is the TVI: a current baseline, scale, target, guardrail, named owner and review date for the outcome. Its artefact is a target card or shared record that the team reads weekly. It prevents a pile of delivery measures from standing in for an outcome.

Read EVD and the CES log weekly to see whether the evidence behind current work is changing. Read the TRB with them when risk is material. The monthly artefact is the DVR decision record, supported by IRT thresholds and the decision log: what the team continued, changed, stopped or scaled, why, and when it will be reviewed. At

quarter end, use DDA and COM to test whether the decision process and outcome measures remain connected to value, quality and risk.

One thing to do this week

Choose one team and one decision due in the next month. Before its next planning meeting, write a one-page TVI: the outcome, baseline, target, guardrail, owner and review date. Ask every attendee to explain it without reading the page. Where their accounts differ, do not move to the work list yet. Resolve the target first, then schedule a fifteen-minute weekly EVD review and name the decision owner for the month-end choice.