ENGINEERING MANAGEMENT ACADEMY

Engineering Management Academy Body of Knowledge

The study resource behind the EMA-I, EMP-II and EME-III certifications · EMA Standard v3.0

EMA Body of Knowledge v3.1 · EMA Competency Framework v3.0

Generated 22 October 2026 · engineeringmanagement.academy/body-of-knowledge

Licensed CC BY-NC-ND 4.0 · Free to share with attribution · Not for commercial use or modification

Contents

About this document 4

EMA Competency Framework v3.0 8

How Judgment Scales 15

People Leadership 23

- PL-1 Delegation, ownership, and decision rights 25
- PL-2 Feedback and difficult conversations 33
- PL-3 Hiring and onboarding 41
- PL-4 Performance management 49
- PL-5 Career development and growth paths 57
- PL-6 Leading through leaders 66

Delivery & Execution 75

- DE-1 Prioritisation under constraint 77
- DE-2 Estimation, planning, and honest commitments 85
- DE-3 Reliability, incident response, and operational ownership 94
- DE-4 Managing scope, risk, and dependencies across workstreams 102
- DE-5 Designing how work flows 110

Strategy & Vision 119

- SV-1 Engineering strategy 121
- SV-2 Communicating and partnering across functions 129
- SV-3 Roadmapping 137
- SV-4 Engineering economics 145
- SV-5 Engineering metrics and measurement 153
- SV-6 Organisational design and leading change 162

Technical Judgment 171

- TJ-1 Evaluating architectural decisions and their organisational consequences 173
- TJ-2 Technical debt 182
- TJ-3 Code review and quality standards 191

TJ-4 Technical credibility and intervention 200
TJ-5 AI-assisted engineering 209
TJ-6 Security, privacy, and compliance as engineering responsibilities 218

Culture & Coaching 226
CC-1 Psychological safety and productive conflict 228
CC-2 Coaching versus directing 237
CC-3 Inclusive practices and equitable opportunity distribution 246
CC-4 Motivation, recognition, and sustainable pace 255
CC-5 Values, integrity, and ethical judgment under pressure 264
CC-6 Self-management 273

Distributed and Hybrid Work 282

Techniques and Models 286
Leading people 287
Deciding 295
Running delivery 296
Strategy and investment 304
Technical stewardship 310
Shaping culture 317

Glossary, A–I 325
Glossary, I–W 330

Sources 335

Changelog 341

Governance 349

How to Cite 352

EMA Body of Knowledge

About this document

The Engineering Management Academy Body of Knowledge is the study resource behind the Academy's three certifications: EMA-I Associate, EMP-II Professional and EME-III Expert. It is one of three documents that together make up EMA Standard v3.0. The Competency Framework says what is examined, this document teaches it, and the Exam Content Outlines say how each examination samples it. A major version revises all three together. A minor version may revise one of them alone, as v3.1 revises this document.

This document follows the framework's organisation exactly: five domains, twenty-nine competencies, one knowledge unit per competency. Every unit has the same structure, so once you have read one you know where to find anything in the others. Each covers what the competency is and why it matters, its core principles, the models and evidence behind it (each model with its evidence grade), its practices, a note on how it scales, its judgment, its tensions, one worked scenario, related competencies and a self-check. The judgment section puts the signals of strong practice and the failure modes in a single list. Two chapters sit alongside the units and are not examined directly. How Judgment Scales is the one place that treats how a competency changes with the scope at which it is exercised. Distributed and Hybrid Work describes the conditions under which most of the competencies are now exercised.

It is deliberately not a collection of facts to memorise. All three examinations test judgment, meaning the ability to read a realistic situation and choose the most defensible action, and the document is written to match. It shows what strong and weak judgment look like at every scope, in situations a manager will recognise.

The three certifications

All three draw on the same twenty-nine competencies (EMA-I examines twenty-seven of them; PL-6 and SV-6 begin at EMP-II). They differ in the scope at which judgment is exercised (one team; several teams through other managers; an engineering organisation through systems and other leaders) and in the examination's length and pass standard. The certifications are cumulative. A Professional is expected to meet every Associate indicator as well as the Professional ones, and an Expert all three.

Level	EMA-I Associate	EMP-II Professional	EME-III Expert
Scope of judgment	one team, direct reports		an engineering organisation, largely

		several teams, usually through other managers	through systems and other leaders
Typical role	first-line engineering manager or team lead	manager of managers, senior manager, group lead	director, vice-president, head of engineering
Questions · time · pass standard	60 · 90 minutes · 70%	75 · 105 minutes · 75%	90 · 120 minutes · 80%
Exam Content Outline	complete	draft; examination not yet open	draft; examination not yet open

Each examination draws an equal number of questions from each of the five domains. The distribution within a domain is stated in that certification's Exam Content Outline.

How to use this Body of Knowledge

- **Read the framework and How Judgment Scales first.** The framework is the map. The scaling chapter explains the scope tags (`[team]`, `[org]`, `[exec]`) that appear throughout the units.
- **Read your certification's Exam Content Outline, then the units in its weight order.** The outline says how many questions each competency carries at your level and what a certified holder can do. Spend most of your time on the highest-weighted units and, within each, on the Judgment and Tensions sections, since scenario questions are built from those.
- **Read the scope below yours as well as your own.** Every scope builds on the one beneath it, and the higher examinations assume the lower judgment is already sound. Executive scenarios often turn on spotting which team-scope instinct no longer applies.
- **Work the self-checks.** Each unit ends with questions at mixed scope, with the answers collapsed. If you can answer the ones at your scope in your own words, you understand the unit.
- **Notice the evidence grades.** Every named model carries `[research]`, `[practice]` or `[contested]`. The grade tells you how much weight the model can bear. A `[contested]` grade means the standard's honest position is that the evidence disagrees.
-

> **Do not confuse familiarity with mastery.** Most candidates recognise the right behaviour and still choose the comfortable one under pressure. The examinations reward the right call, especially when it is the hard one.

A note on the models cited

The units cite widely recognised models and research from the engineering-leadership canon and from the organisational and psychological literature. Situational leadership, the situation-behaviour-impact feedback structure, the DORA delivery metrics, psychological safety and the planning fallacy are among many. Treat them as tools for reasoning; they are not doctrine, and no examination question asks you to name a framework. Every model carries an evidence grade, and every source is a real, checkable work listed in the Sources chapter. The text is original throughout. The standard cites its sources and does not reproduce them.

Version and changelog

This is version 3.1 of the Body of Knowledge: the text released with EMA Standard v3.0 on 26 September 2026, with its prose made plainer on 7 October 2026 and, on 8 October, corrected where the rewrite had moved a meaning or a source had been misattributed (see the changelog). The changelog records what changed from the previous edition and why, including the move from a separate version line for each document to one major version for the whole standard, with minor versions numbered per document. The previous edition remains readable, unchanged, in the archive, and a banner on each archived page links to its current equivalent.

Licence

The EMA Body of Knowledge and the Exam Content Outlines are published under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 licence, except for two parts. You may copy, share, print and cite them, with attribution, for non-commercial purposes, and without modification. To translate them, adapt them, or teach from them commercially, ask. The two exceptions are published under the Creative Commons Attribution 4.0 licence, so that they can be adopted and adapted freely, with attribution, in job descriptions, career ladders, rubrics and tools. They are the competency list of the EMA Competency Framework (the twenty-nine identifiers, titles and one-line descriptions) and the proficiency indicators of the Exam Content Outlines (each statement of what a certified holder can do). The examination questions are not published and are licensed under neither. How to Cite gives the citation forms and the attribution line. Neither licence grants any

right to use the names Engineering Management Academy, EMA-I, EMP-II or EME-III, which remain the Academy's marks.

EMA Competency Framework v3.0

What this framework is

Every Engineering Management Academy examination is built against this framework. It names the twenty-nine competencies an engineering manager is assessed on, grouped into five domains, and it is published in full so that a candidate knows exactly what is examined and an employer knows exactly what a credential means.

The framework is a **standard**, not a syllabus. It says what capability looks like; it does not say how to acquire it. The Academy sells no training and endorses none, which is what allows the standard to be independent of any course, and why the credential certifies capability rather than attendance.

Three documents make up EMA Standard v3.0 and are released together:

- **This framework**: the competencies and their permanent IDs.
- **The EMA Body of Knowledge**: one knowledge unit per competency, written for judgment rather than recall, plus two cross-cutting chapters.
- **The Exam Content Outlines**: one per certification, stating for every competency how many questions it carries, what a certified holder can do, and what kinds of situations the questions present.

The five domains

Every examination weights the five domains equally: 20% each, which is 12 questions of 60 on EMA-I, 15 of 75 on EMP-II, and 18 of 90 on EME-III. The balance is deliberate. Engineering management fails at its weakest dimension; a brilliant strategist who cannot give feedback, or a beloved coach who cannot ship, is not yet the complete manager this credential certifies.

Within a domain, the questions are not spread evenly across competencies. Each certification's Exam Content Outline states the distribution for that level, and it changes with level: a competency that carries no questions on EMA-I may carry a third of its domain on EME-III.

Domain 1: People Leadership

Building, growing and retaining effective engineers, teams and, at scale, the leaders of teams.

ID	Competency	One line
PL-1	Delegation, ownership, and decision rights	matching scope to capability, making authority explicit, not taking work back
PL-2	Feedback and difficult conversations	timely, specific, behaviour-anchored, in both directions
PL-3	Hiring and onboarding	structured evaluation, calibrated standards, assessing capability rather than tool access
PL-4	Performance management	expectations, support, honest consequences, and fair process
PL-5	Career development and growth paths	one-on-ones, promotions, the Staff+ track, and the IC/manager choice
PL-6	Leading through leaders	selecting, developing, and holding accountable the managers who report to you

Domain 2: Delivery & Execution

Turning intent into shipped, reliable software, predictably, under constraints that never go away.

ID	Competency	One line
DE-1	Prioritisation under constraint	sequencing, trade-offs, saying no
DE-2	Estimation, planning, and honest commitments	managing uncertainty rather than chasing accuracy
DE-3	Reliability, incident response, and operational ownership	on-call, SLOs, postmortems, learning from failure
DE-4	Managing scope, risk, and dependencies across workstreams	seeing the whole, naming the risk, negotiating the boundary
DE-5	Designing how work flows	

		process selection, developer experience, and AI-assisted workflows

Domain 3: Strategy & Vision

Connecting engineering effort to business outcomes and direction, and shaping the organisation that does the work.

ID	Competency	One line
SV-1	Engineering strategy	translating business goals into technical direction, and writing it down
SV-2	Communicating and partnering across functions	product, design, executives, and non-engineering stakeholders
SV-3	Roadmapping	balancing product delivery against platform and quality investment
SV-4	Engineering economics	headcount, budget, cost of delivery, vendors, and build/buy/AI decisions
SV-5	Engineering metrics and measurement	choosing, interpreting, and not gaming them, including measuring AI impact
SV-6	Organisational design and leading change	team structures, ownership boundaries, reorgs, and transitions

Domain 4: Technical Judgment

Exercising sound engineering judgment without doing the engineering, and owning the technical responsibilities that cannot be delegated away.

ID	Competency	One line
TJ-1		

	Evaluating architectural decisions and their organisational consequences	trade-offs, reversibility, and who has to coordinate because of this
TJ-2	Technical debt	recognising, quantifying, and scheduling repayment
TJ-3	Code review and quality standards	including standards for AI-generated code
TJ-4	Technical credibility and intervention	staying close enough to judge, and knowing when to step in
TJ-5	AI-assisted engineering	adoption, governance, capability-building, and risk
TJ-6	Security, privacy, and compliance as engineering responsibilities	owned by engineering, not delegated to a function

Domain 5: Culture & Coaching

Shaping the environment in which good engineering happens, developing the judgment of the people in it, and sustaining the manager who does both.

ID	Competency	One line
CC-1	Psychological safety and productive conflict	safe to speak, expected to disagree, committed once decided
CC-2	Coaching versus directing	developing judgment in others
CC-3	Inclusive practices and equitable opportunity distribution	who gets the stretch work, the visibility, and the benefit of the doubt
CC-4	Motivation, recognition, and sustainable pace	including in distributed and hybrid teams
CC-5	Values, integrity, and ethical judgment under pressure	what you do when the right call is the costly one
CC-6	Self-management	the manager's own time, energy, boundaries, and resilience

How the three certifications use one framework

All three certifications draw on the same twenty-nine competencies. EMA-I examines twenty-seven of them: leading through other leaders (PL-6) and organisational design (SV-6) begin at EMP-II. What changes between them is not the list but the **scope** at which judgment is exercised: the object being decided about, the time horizon, the ambiguity, the consequence of error, who the manager acts through, and how late the information arrives. An Associate delegates a project; a Professional delegates a team; an Expert delegates the design of the organisation. The competency is the same. The judgment is not.

The Body of Knowledge treats this once, in the chapter *How Judgment Scales*, rather than three times per competency. Each knowledge unit carries a short scaling note and tags its signals of strong and weak practice with the scope where they apply (`[team]`, `[org]`, `[exec]`) and each Exam Content Outline states, per competency, what a certified holder at that level can do.

Certification	Scope of judgment	Typical role	Questions · time · pass standard
EMA-I Associate	one team, direct reports	first-line engineering manager or team lead	60 · 90 minutes · 70%

EMP-II Professional	several teams, usually through other managers	manager of managers, senior manager, group lead	75 · 105 minutes · 75%
EME-III Expert	an engineering organisation, largely through systems and other leaders	director, VP, head of engineering	90 · 120 minutes · 80%

The certifications are cumulative. A Professional is expected to meet every Associate indicator as well as the Professional ones; an Expert, all three.

Why these twenty-nine

The framework is deliberately not a list of everything a manager does. Each competency earns its place by being a distinct **judgment** (a kind of decision with its own failure modes, its own tensions and its own evidence) that a scenario question can test. Where two competencies turned out to be one judgment, they were merged (TJ-4). Where one competency was carrying two, it was split (PL-6 and SV-6 out of the old delegation and performance material). Where the job had changed since v1.0 and no competency covered the change, one was added (TJ-5, TJ-6, CC-6).

The cross-cutting chapters (*How Judgment Scales* and *Distributed and Hybrid Work*-) are not competencies and are not examined directly. They are referenced from the competencies they touch.

About this version

v3.0, published with EMA Standard v3.0. The previous framework was v1.0 (June 2026); the Body of Knowledge was v2.0. From this release the framework, the Body of Knowledge and the Exam Content Outlines are versioned as one standard: a major version revises all three together and they share its number. A minor version revises one document and carries its number only there, as Body of Knowledge v3.1 does; this framework stays at v3.0. There is no framework v2.0; the number was skipped so that "v3.0" means the same release wherever it appears.

What changed from v1.0. Five domains and their equal weighting are unchanged, as are the five indexed domain pages. The twenty-five competencies of v1.0 became twenty-nine: twenty-three continue with the same scope or a wider one, two merged into one, and five are new. Every change is listed, with its reason, in the changelog.

What it means for existing credentials. A credential records the framework version it was examined against; every credential issued before the v3.0 effective date carries `1.0`, and the public verification page and API show it. Nothing about an existing credential changes. Examinations continue to draw from the same five domains at the same weights; the per-competency distributions in the Exam Content Outlines describe the standard the question bank is being developed toward, and the outlines say so plainly.

Governance. The framework is revised on a stated cadence (a major version at most annually, minor corrections quarterly) and a major revision is preceded by a public comment period. Development of examination questions is kept separate from development of the Body of Knowledge, so that the study material can never quietly become the answer key.

How Judgment Scales

Why one chapter instead of seventy-five bands

The three certifications draw on the same twenty-nine competencies, and EMA-I examines twenty-seven of them (PL-6 and SV-6 begin at EMP-II). An Associate, a Professional and an Expert hold the same competencies; what separates them is the scope at which each one is exercised. The previous edition of this standard expressed that by writing every competency three times, once per level. Readers found the bands repetitive, and the repetition hid the real pattern, which is that scope changes every competency in the same handful of ways. This chapter states those ways once. Each knowledge unit then carries a short scaling note and tags its signals of strong and weak practice with the scope where they apply, and the Exam Content Outlines say what each level's questions expect.

Read this chapter before the domain units. It teaches the pattern every unit follows, and it is the key to the scope tags.

The six dimensions of scope

Scope is neither seniority nor headcount. A manager with thirty reports in one team is still exercising team-scope judgment; a director with twelve people across three teams is not. Scope is what changes about a decision when the object of the decision gets larger. Six things change, and they change together.

1. The object of judgment

The most useful question to ask of any competency is *what is being decided about?* At team scope the object is concrete and close: a task, an engineer, a sprint, a service, a conversation. At organisational scope it is a class of those things, such as a team's charter, a manager's judgment, a delivery system, a platform or a hiring bar. At executive scope it is the system that produces the class, whether that is the structure that decides who owns what, the leadership pipeline, the operating cadence, the technical strategy or the culture. The table in the next section gives the object for every competency. Of everything about how a competency scales, this matters most, because applying the same instinct to the wrong object is the commonest way strong managers fail after promotion.

2. Time horizon

Team-scope decisions resolve in days or weeks, and their feedback arrives before the next similar decision. Organisational decisions play out over quarters: a manager appointment,

a team split, a platform bet. Executive decisions play out over years. Some of the most consequential, such as a technical strategy, an organisational design or a culture, are judged only in hindsight, by people who will not know what the alternatives were. The longer the horizon, the less a decision can be corrected by iteration and the more it has to be right on first principles.

3. Ambiguity

At team scope the goal is usually given, and the question is how to reach it. At organisational scope the goals of different teams conflict, and the question is which to favour. At executive scope the goal itself is often unclear, contested or shifting, and part of the work is deciding what the question is. Strong judgment at each level is comfortable with the ambiguity that level carries. It does not manufacture false certainty to escape it, and it does not import ambiguity the level does not have. An Associate who treats a clear sprint goal as a strategic question is as mis-scoped as an Expert who treats a strategic question as a scheduling problem.

4. Consequence and reversibility

A bad delegation costs a week and teaches everyone something. A bad manager appointment costs a team a year. A bad organisational design costs a company the ability to execute for as long as it stands. As scope widens, the cost of error rises and the chance to reverse falls, so deliberation belongs where the door is one-way. The opposite failure exists as well. Executives who bring executive-grade deliberation to every reversible decision slow the whole organisation to their own pace.

5. Who you act through

At team scope you act directly. You have the conversation, review the design and make the call. At organisational scope you act mostly through the managers who report to you, and your leverage comes from what you delegate to them, how you develop them and what you hold them to. At executive scope you act largely through systems (structures, cadences, standards, incentives) and through leaders you rarely see day to day. Each step takes you further from the work and adds a layer whose judgment you have to develop and trust. Every competency in this standard has a "through others" form. At every level, the newly promoted are most recognisable by the temptation to skip the layer and act directly.

6. Information latency and filtering

The further you are from the work, the later and more filtered your information arrives. A team lead hears about a problem the day it happens. A director hears a version of it in a weekly update, after the manager has decided how to present it. An executive hears a narrative weeks later, shaped by everyone it passed through. Strong judgment at scope

compensates on purpose, with skip-level conversations, direct observation of artifacts, metrics that do not pass through a person, and a visible reaction to bad news that makes it safe to deliver early. Weak judgment at scope hears no bad news and concludes there is none.

The object of judgment, competency by competency

Each unit's scaling note points to the table below. For each competency it names what is being decided about at each of the three scopes. Read down a column to see a level, or across a row to see how one competency changes shape.

ID	Competency	Team	Organisation	Executive
PL-1	Delegation, ownership, and decision rights	a task or project, and the authority that goes with it	a team's charter, and a manager's decision rights	the organisation's decision-rights design: who decides what, at which level
PL-2	Feedback and difficult conversations	an engineer's behaviour and its impact	a manager's leadership, and feedback that crosses team boundaries	a leader's performance, and the feedback culture the organisation rewards
PL-3	Hiring and onboarding	an individual hire and their first quarter	a team's hiring bar, and the appointment of managers	the hiring system: calibration across the organisation, senior and executive appointments
PL-4	Performance management	an engineer's performance against clear expectations	a manager's performance, including the performance of their team	the performance system: fairness, consistency and consequences across the organisation
PL-5			the growth paths available across	the career architecture: levels, tracks, and

	Career development and growth paths	an individual's growth and next step	teams, and who gets them	the leadership pipeline
PL-6	Leading through leaders	not exercised at team scope	the managers who report to you: selection, development, accountability	the leadership team as a system, and the leaders of leaders
DE-1	Prioritisation under constraint	the team's queue	competing priorities across teams, and the shared bottleneck	the portfolio: which bets the organisation makes and which it declines
DE-2	Estimation, planning, and honest commitments	the team's forecast and its commitment	commitments that depend on several teams, and the truthfulness of rolled-up status	the organisation's planning cadence and the credibility of its commitments to the business
DE-3	Reliability, incident response, and operational ownership	an incident in progress, and one service's operational health	reliability across services, ownership gaps, and the on-call system	reliability as a strategic property: investment, risk appetite, and the resilience the business assumes
DE-4	Managing scope, risk, and dependencies	one workstream's scope and risks	dependencies between teams, and risks that fall between them	systemic risk: the dependencies the organisation does not see, and the programmes that span it
DE-5			consistency and autonomy across teams; the	the organisation's operating system: cadence, standards, and the investment in

HOW JUDGMENT SCALES

	Designing how work flows	the team's process and tools	developer platform	developer experience
SV-1	Engineering strategy	translating a given strategy into the team's work	a department strategy that connects several teams to business outcomes	the engineering strategy itself, argued at the executive table
SV-2	Communicating and partnering across functions	the team's product and design partners; a stakeholder pressing for a date	department-level partners; executives who need trade-offs explained	the executive team and the board; engineering's standing in the company
SV-3	Roadmapping	the team's roadmap and its platform debt	a department roadmap balancing several teams' product and platform work	multi-year investment across product, platform and quality
SV-4	Engineering economics	a build-versus-buy call for the team; the team's cost	a department budget and headcount plan; vendor decisions	the economics of engineering as a whole: cost of delivery, capital allocation, make/buy/AI at scale
SV-5	Engineering metrics and measurement	the team's metrics and what they hide	metrics that compare or aggregate teams, and the gaming they invite	the measures the business uses to judge engineering, including AI's effect on them
SV-6	Organisational design and leading change	not exercised at team scope	team boundaries and ownership within a department; a reorg of several teams	the shape of the organisation: structure, transitions, mergers, and the change that follows

HOW JUDGMENT SCALES

TJ-1	Evaluating architectural decisions	one team's design decisions and their reversibility	architecture across teams: standards at boundaries, platform bets	the technical strategy: the bet-the-company decisions and their organisational consequences
TJ-2	Technical debt	the team's debt and its repayment schedule	systemic debt that no single team owns; department-level investment	debt as a strategic liability: the platform's runway and the business's exposure
TJ-3	Code review and quality standards	the team's review practice and quality bar	consistent standards across autonomous teams, including for AI-generated code	the organisation's quality posture and the systems that maintain it
TJ-4	Technical credibility and intervention	when to step into a design or a fix, and when not to	staying close enough to several teams' work to judge it without doing it	technical credibility at executive level: what to track personally, and when to intervene at all
TJ-5	AI-assisted engineering	how the team uses AI assistance, and the review it needs	adoption and governance across teams; capability-building	the organisation's AI strategy for engineering: investment, risk, and what changes about the work
TJ-6	Security, privacy, and compliance	the team's security practice and a vulnerability in its service	a class of gap across several teams; ownership of the fix	security, privacy and compliance as organisational risk, owned at executive level
CC-1				

	Psychological safety and productive conflict	safety in the room: how you react to bad news and dissent	safety between teams and in the managers you lead; conflict between teams	trust in leadership across the organisation, and what leadership visibly does
CC-2	Coaching versus directing	developing an engineer's judgment rather than supplying the answer	coaching the coaches: developing managers' judgment	developing leaders' judgment, and a leadership culture that coaches
CC-3	Inclusive practices and equitable opportunity	who on the team gets the stretch work and the visibility	equity across teams: promotion and opportunity patterns	the organisation's opportunity structure and the data that reveals it
CC-4	Motivation, recognition, and sustainable pace	one team's motivation and pace, in the room or distributed	motivation and pace across teams; recognition that crosses boundaries	the organisation's relationship to pace: what leadership rewards, and what it survives
CC-5	Values, integrity, and ethical judgment	the costly right call in front of the team	holding managers to the values; the brilliant-jerk decision at department scale	values under company-level pressure: what leadership does when it is expensive
CC-6	Self-management	your own time, energy and boundaries as a first-line manager	your own effectiveness as the layer between teams and executives	your own resilience and boundaries under executive-scale consequence and visibility

Two competencies, PL-6 and SV-6, have no team-scope object. Leading through leaders requires leaders to lead through, and organisational design requires an organisation to design. Both are therefore assessed from EMP-II upward, and the EMA-I Exam Content Outline lists them explicitly as not assessed.

Reading the scope tags

Each unit's judgment section puts signals of strong practice and failure modes in one list, and tags an item with the scope where it applies:

- `[team]`: one team, direct reports; the manager acts directly.
- `[org]`: several teams, usually through other managers.
- `[exec]`: an engineering organisation, largely through systems and other leaders.

An item with no tag applies at every scope. An item tagged `[org] [exec]` applies at both of those and not at team scope. The tags map onto the certifications: EMA-I assesses `[team]`, EMP-II adds `[org]` and EME-III adds `[exec]`. Even so, they describe the judgment, whoever is reading. A first-line manager reading an `[exec]` item is reading about the scope at which their own manager's manager works. That helps in understanding the decisions that arrive from above, and in knowing what the next level of the job involves.

The Exam Content Outlines use the same three words as the `scope` of every proficiency indicator. Because the certifications are cumulative, a Professional is held to every `team` indicator as well as every `org` one.

What does not scale

Some things stay the same at every scope, and they are worth naming. The obligation to tell the truth about status does not become more negotiable with seniority; if anything, the consequence of shading it grows. The distinction between an estimate and a commitment is the same at every level. The reaction to bad news that makes it safe to deliver is the same reaction whether the messenger is an engineer or a vice-president. And the question that tests any plan works at every altitude: pick any item and ask the person doing it why it is there.

People Leadership

What this domain is

People Leadership is the work of building, growing and retaining effective engineers and teams and, as scope widens, the leaders of teams. It is the most human part of the job. At every scope it is also where the strengths of the previous scope turn into liabilities. A strong engineer's instinct is to solve the problem personally, which is nearly the opposite of what makes a strong team leader. A strong team leader's instinct is to stay close to every person and every problem, which is nearly the opposite of what makes a strong leader of managers.

The domain carries 20% of every examination: 12 questions on EMA-I, 15 on EMP-II and 18 on EME-III. Its six competencies are listed below with the questions each carries at each level, and the Exam Content Outlines give the proficiency indicators.

The throughline

Each competency in this domain is judgment about people under uncertainty. Whom do you trust with what? What feedback do you give, and when? Whom do you hire and promote, how do you handle underperformance, how do you invest in growth, and, from organisational scope, who leads? None of these has a formulaic answer. The examinations reward the response that balances care for the individual against responsibility to the team, the organisation and the work. They are built from the familiar ways that balance fails: the work taken back, the conversation avoided, the bar lowered, the plan issued without the expectation, the promotion given to retain someone, the manager reached around.

How the domain scales

This domain's arc is the clearest in the framework. At team scope you lead people. At organisational scope you lead the people who lead people, which is a different job with its own competency, PL-6 Leading through leaders. At executive scope you build the system that produces and sustains leaders: the career architecture, the leadership pipeline, the performance and hiring systems, and a design of decision rights that keeps most decisions from ever reaching you. The general pattern is in How Judgment Scales; each unit's scaling note places its own competency on it.

Reading order

Start with PL-1 Delegation, ownership, and decision rights. The other five depend on it, and it is the reference unit for the template the whole standard uses. PL-2 Feedback and difficult conversations and PL-4 Performance management are a pair: the feedback that comes before any performance conversation, and what happens once feedback has failed to change the behaviour. PL-3 Hiring and onboarding and PL-5 Career development and growth paths cover the two ends of a person's time with the organisation. Read PL-6 Leading through leaders last. It assumes all five and is assessed from EMP-II.

Across domains

The people competencies depend on Culture & Coaching more than on any other domain. CC-1 Psychological safety and productive conflict supplies the safety that decides whether feedback is given at all, CC-2 Coaching versus directing offers coaching as the alternative to rescue, CC-3 Inclusive practices and equitable opportunity distribution governs who gets the opportunity, and CC-5 Values, integrity, and ethical judgment under pressure covers the costly right call about a valued person. Designing the structure people sit in is SV-6 Organisational design and leading change, and knowing which technical decisions to keep is TJ-4 Technical credibility and intervention.

PL-1 Delegation, ownership, and decision rights

matching scope to capability, making authority explicit, not taking work back

1. Definition and why it matters

Delegation is how a manager's output stops being limited by their own hours, and how the people around them grow into work they have not done before. It involves three linked acts. The manager hands someone an outcome rather than a set of steps, gives them the authority to reach it, and leaves the work with them when it gets hard. Ownership is what the person on the receiving end has once all three have happened. Decision rights are the explicit statement of who may decide what; most managers leave them implicit, and most organisations pay for that. The competency matters because almost every other one depends on it. A manager who cannot delegate cannot coach, cannot lead through other managers, cannot make time for strategy and cannot be away for a week. The examinations test it because its failure modes are so recognisable: the task handed over and then quietly taken back, the authority granted and then overridden, the decision nobody knew was theirs to make.

2. Core principles

1. **Delegate the outcome, and be explicit about three things.** A delegation is complete when the person knows the result wanted, the constraints they are working within (time, budget, what not to touch), and the level of authority they hold: decide and proceed, decide and tell me, or recommend and check. Leaving any of the three unsaid leaves a gap the person will fill by guessing, and that is no form of trust.
2. **Match scope to capability on this kind of work, plus a deliberate stretch.** The guide is the person's experience with this specific task, whatever their seniority. Give enough challenge that they grow, and stop short of the point where they fail without support.
3. **Delegated authority is real or it is nothing.** If you grant someone the right to decide and then quietly reverse them, you have taught them, and everyone watching, that the grant was theatre. Overrule rarely, openly, and with the reason.
4. **Accountability does not transfer.** You hand over the work and the decisions and keep the answerability for the result. That is what separates delegation from abdication, and it holds whether the thing delegated is a task, a team or a division.

5. **Do not take it back.** When delegated work wobbles, the manager can support, coach or rescue. Rescue feels responsible and is usually the mistake. It ends the growth and teaches the team that ownership is provisional.
6. **Decision rights are designed, not discovered.** Wherever more than one person could reasonably think a decision is theirs, write down whose it is before the disagreement arrives rather than in the middle of it.

---

3. Models and evidence

Delegation is among the most written-about subjects in management and among the least studied empirically. The models below are practitioner models with clear provenance. None of them has the large-sample evidence that sits behind psychological safety or the DORA findings, and this unit does not pretend otherwise.

Task-relevant maturity [practice]

In Andrew S. Grove, High Output Management (1983), Andrew Grove observes that how much direction a person needs depends on their experience with this particular kind of work rather than on their general seniority. A senior engineer commanding their first incident has low task-relevant maturity at incident command and needs close support there, while needing none on system design. A strong engineer newly promoted into management has low task-relevant maturity at management itself, however senior they are technically. In practice, delegation scope and support level are set per task and per person and reassessed as the work changes, because yesterday's calibration drifts into today's mismatch without anyone noticing.

Situational leadership [practice]

This model grew from Paul Hersey and Kenneth H. Blanchard, Management of Organizational Behavior: Utilizing Human Resources (1969). It treats leadership style as a variable, flexed between directing, coaching, supporting and delegating according to the person's competence and confidence on the task in front of them. Its lasting insight is that a manager's style should follow the task, and should not be set by the manager's personality or the report's title. Misread it one way and you micromanage experts; misread it the other and you abandon novices. The model has been reworked many times since it was first published and its predictive claims are thin. This unit uses it as a vocabulary for calibrating support and makes no use of it as a theory.

Levels of authority and intent-based leadership [practice]

The most useful single mechanism for making authority explicit comes from L. David Marquet, Turn the Ship Around!: A True Story of Turning Followers into Leaders (2012). The person stops asking "may I..." and starts saying "I intend to...", stating what they are about to do and why. The manager's silence counts as consent, and an objection has to be spoken. Decisions move to where the information is, the manager stays informed, and the level of authority shows in every exchange. Formalised, the same idea becomes a ladder of authority levels agreed for each kind of decision: recommend, decide and inform, decide alone.

One-way and two-way doors [practice]

Jeff Bezos sorts decisions into two kinds in Jeff Bezos, 2015 Letter to Shareholders (Amazon.com, Inc.) (2016). Most are reversible two-way doors and should be made quickly by the people closest to them. A few are one-way doors that deserve deliberation and senior involvement. For delegation, this answers the question every manager asks about which decisions can be handed over. Most of them can. The manager's job is to name the few one-way doors and keep those, and to stop hovering over all the rest. Reversibility is often misjudged, so when in doubt treat a door as one-way until it is shown to be otherwise.

Managerial leverage [practice]

Grove also argues, in Andrew S. Grove, High Output Management (1983), that a manager's output is the output of the organisation under them, and that some activities return far more of that output than others. An hour spent making a delegation clear pays back across every hour the delegate then spends on it. This is the economic case for the explicitness in principle 1. The brief that feels slow to write is the part of the delegation that returns the most.

---

4. Practice

The delegation brief

For anything larger than a day's work, the delegation is written down, even if only as a paragraph in a message. It states the outcome and how it will be recognised, the constraints, the authority level, who else is involved and in what role, and when the two of you will next look at it together. Writing it exposes what the manager has not yet decided. If the brief cannot be written, the delegation is not ready.

A decision-rights register

This is a short, living list of the team's recurring decisions (releases, architectural choices above a certain size, hiring decisions, on-call changes, scope trades) with who decides each and at what authority level. Most teams have never written one down, and they are surprised by how many entries turn out to be disputed. The register also makes principle 3 checkable, since an override of a listed right is visible as an override.

The check-in that is about the outcome

A scheduled conversation about delegated work asks about the outcome and the risks to it, and leaves the steps alone. Compare "What is the state of the migration and what could stop it?" with "Did you talk to the platform team yet?" The first keeps ownership with the delegate. The second takes the steering wheel back one question at a time.

Handing back, explicitly

Sometimes a delegation has to be withdrawn because the person is not ready, the situation has changed or the stakes have risen. Saying so plainly ("I am taking this back, here is why, here is what you keep") protects the person's standing and keeps everyone else believing that delegation means something. Withdrawing it quietly, by fielding the decisions yourself while the title stays with them, destroys both.

Onboarding a new mandate

When someone takes on a larger scope, the first weeks go on making their authority real. The manager routes questions back to them in front of others, declines to answer on their behalf and is seen to accept their decisions. Authority granted in private and undermined in public does not count as authority.

---

5. Scaling note

The thing being delegated changes with scope; the discipline does not. At team scope you delegate tasks and projects to engineers. At organisational scope you delegate whole teams to managers, and the characteristic failure becomes reaching around the manager to their engineers. At executive scope ownership is designed through structure, charters and decision rights instead of being assigned one conversation at a time, and the failure is becoming the decision bottleneck of a hub-and-spoke organisation. Leading the managers you delegate teams to is a competency of its own, PL-6 Leading through leaders, and designing ownership through structure is SV-6 Organisational design and leading change. The general pattern, with the table placing every competency at each scope, is in How Judgment Scales.

---

6. Judgment

- Delegates outcomes and leaves the method to the person. [team]
- Gives a stretch assignment and stays close enough to make it survivable. [team]
- States the authority level in the delegation and honours it afterward.
- Resists the urge to take work back when it wobbles, and chooses support or coaching over rescue. [team]
- Delegates the interesting decisions as well as the routine work. A team that only ever receives the boring tasks learns that delegation is a chore, when it should be an investment. [team]
- Failure mode (the dump): an under-specified task handed over with no brief, no check-in and no authority, followed by disappointment. [team]
- Failure mode (the quiet override): authority granted, then reversed without saying so, usually because the manager would have decided differently. [team] [org]
- Failure mode (the rescue): stepping in at the first sign of struggle. It feels responsible, and it teaches the team that ownership is provisional.
- Hands each manager a real mandate, with the decision rights between the two of you made explicit. [org]
- Uses skip-level conversations for signal, never for decisions that belong to the manager. [org]
- Failure mode (reaching around): managing a manager's engineers directly, fixing their team's problems, or letting their escalations become your workload, because you used to do their job and could do it faster. [org]
- Failure mode (taking the team back): withdrawing a manager's mandate when their team struggles, when the job is to develop the manager (see PL-6 Leading through leaders). [org]
- Delegates large mandates to senior leaders and keeps accountability undiluted. [exec]
- Failure mode (the hub-and-spoke bottleneck): an organisation in which every significant decision still routes through one person, because nobody ever designed the decision rights. [exec]
- Failure mode: mistaking empowered leaders for a state in which nobody is accountable. [exec]

---

7. Tensions

Stretch versus safety. People grow through assignments they have not done before, and every such assignment puts the outcome at some risk. The judgment lies in how big the

stretch is and how strong the net is. A stretch with no safety net is a gamble with someone else's reputation, and a net so tight that nothing is stretched is no longer delegation.

Speed versus development. The manager could often do it faster, and in a real emergency should. The trouble is that "faster this once" compounds. Every task the manager keeps is one the team does not learn, and a manager who is always faster ends up with a team that cannot operate without them.

Authority versus accountability. Handing over the right to decide while keeping the answerability for the result is uncomfortable by design. Ease the discomfort by keeping the decisions and you abdicate the development; ease it by disclaiming the result and you abdicate the job.

Explicitness versus trust. Writing down decision rights and authority levels can feel like bureaucracy, or like a statement that people are not trusted. In practice the written form is what makes trust checkable. The right that gets overridden is the unwritten one, because nobody can point to it.

Autonomy versus consistency. The more decisions go to the people closest to the work, the more those decisions diverge across people and teams. Some of that divergence is the point and some is a cost. Deciding which decisions must be consistent, and holding those centrally, is part of designing decision rights and no exception to it.

8. Worked scenario

A first-line manager has delegated a database migration to an engineer with four years' experience who has never led a project of this size. The brief was reasonable. It gave the outcome, a two-month window, a weekly check-in, and authority to make design decisions within the existing platform. Three weeks in, the check-in shows that the engineer has spent most of the time on a schema redesign the migration did not require, has not yet spoken to the two teams whose services depend on the table, and is visibly anxious. The launch that depends on the migration is five weeks away.

The pull toward rescue is strong. The manager has done three migrations and could re-plan this one in an afternoon. The scope was a stretch and the wobble is real, but the diagnosis has to come before the response. The engineer's task-relevant maturity at leading a multi-team change was low, and the brief did not make up for it. It named the outcome and the authority and said nothing about the dependencies, which is exactly where an engineer who has only worked inside one team will not look. So far the failure lies in the delegation, not in the delegate.

The manager decides to keep the delegation and repair the brief. In the check-in they name what is now known: the schema redesign is out of scope, and the dependent teams are the critical path. They ask the engineer to bring a revised plan by the end of the week that

starts with those conversations. They offer one concrete piece of support where maturity is lowest, sitting in on the first conversation with the dependent teams as an observer while the engineer leads. They do not take over the plan, and they do not say "I'll handle the platform team." Because accountability has not moved, they also make the risk visible upward. The launch owner hears now that the migration is behind and what is being done about it, instead of hearing it in five weeks.

If the revised plan is sound, the engineer finishes a project they could not have finished without the wobble, and the manager gains a second person who can lead a migration. If it is not, the manager withdraws the delegation explicitly, gives the reason, and leaves the engineer a defined part of the work. Either way, the manager does not do the thing that felt most responsible three weeks in.

9. Related competencies

- PL-6 Leading through leaders: the managers you delegate whole teams to, and what developing them requires. This is the organisational form of the competency.
- SV-6 Organisational design and leading change: ownership designed through structure, charters and reorganisation, which is the executive form.
- CC-2 Coaching versus directing: the coaching that replaces rescue when delegated work wobbles.
- PL-5 Career development and growth paths: delegation as the main instrument of growth, and the stretch assignment as a career move.
- TJ-4 Technical credibility and intervention: deciding which technical decisions are yours to keep and which to hand over.

10. Self-check

1. An explicit delegation makes three things clear. What are they, and which one do managers most often leave out? *Answer:* The outcome wanted, the constraints, and the level of authority. Authority is the one most often left implicit. Its absence produces both the delegate who checks everything and the delegate who oversteps.
2. Why is task-relevant maturity a better guide to delegation scope than seniority? *Answer:* The support a person needs depends on their experience with this kind of work. A senior engineer can have low maturity at a task they have never done, and a junior one can be highly capable at a task they have done many times. Seniority predicts neither.
3. Delegated work is wobbling three weeks in. Name the three responses available and say which one usually feels most responsible and is usually wrong. *Answer:* Support,

coach, or rescue. Rescue feels most responsible and is usually wrong, because it ends the person's growth and teaches the team that ownership is provisional. The exception is a real emergency, and even then the withdrawal should be explicit.

4. You granted an engineer authority to decide the design of a feature and you disagree with what they chose. What are your options and what does each teach the team? *Answer:* You can accept it, which teaches that authority is real. You can overrule it openly with the reason, which teaches that authority is real but bounded. Or you can reverse it quietly, which teaches that the grant was theatre. Avoid the third, and keep the second rare.
5. A manager who reports to you is drowning. What distinguishes coaching them from quietly doing their job? [org] *Answer:* Coaching leaves the decisions and the team with the manager and develops their judgment about both. Doing their job takes the decisions back and leaves them only the title. The test is who is deciding at the end of the week.
6. What does "I intend to..." change about a delegation compared with "may I..."? *Answer:* The decision moves to the person with the information and the manager stays informed. The authority level also becomes visible in every exchange, since silence is consent and an objection has to be spoken.
7. In an organisation you have just inherited, every significant decision still reaches you. What is the diagnosis, and what is the first repair? [exec] *Answer:* Nobody designed the decision rights, so the structure routes everything to the hub. The first repair is to write down which recurring decisions are made where, push the reversible ones down explicitly and keep only the one-way doors. Working faster at the hub is no repair.

Sources

- Andrew S. Grove, High Output Management (1983): task-relevant maturity; managerial leverage.
- Paul Hersey and Kenneth H. Blanchard, Management of Organizational Behavior: Utilizing Human Resources (1969): the origin of situational leadership.
- L. David Marquet, Turn the Ship Around!: A True Story of Turning Followers into Leaders (2012): intent-based leadership and "I intend to".
- Jeff Bezos, 2015 Letter to Shareholders (Amazon.com, Inc.) (2016): one-way and two-way doors.
- Will Larson, An Elegant Puzzle: Systems of Engineering Management (2019): sizing mandates and spans; delegation as a system property at scale.

PL-2 Feedback and difficult conversations

timely, specific, behaviour-anchored, in both directions

1. Definition and why it matters

Feedback is how behaviour changes, and a difficult conversation is feedback at the moment it costs something to give. When feedback fails, it is usually not because it is wrong. The delivery is the problem: too vague to act on, too late to matter, or so cushioned that the message disappears. The competency covers both directions. A manager who gives feedback well and never asks for it has taught the team that information only flows downward, and has cut the channel that would tell them when their own judgment is failing. Feedback is the cheapest corrective a manager has and the one most often put off. The conversation avoided in week two becomes the performance problem in quarter three and the resignation in quarter four, and the rest of the team quietly learns that the pattern is tolerated. The examinations test it because avoidance is the most recognisable mistake in management, and because at every scope the wrong option is usually the comfortable one.

2. Core principles

1. **Anchor feedback in a situation, a behaviour and an impact.** Say what happened, what the person observably did and what it caused. Leave out what they are and why you think they did it: diagnosing character or intent produces defensiveness, and the diagnosis is usually wrong as well.
2. **Early and small beats late and large.** Frequent small corrections make feedback ordinary, so it stops arriving as a verdict. Feedback saved for the review cycle comes too late to change the quarter it describes and too heavy to be heard.
3. **Clarity is the kindness.** Softening a hard message until nobody could recognise it passes your discomfort to the person, who will find out the real message later and at higher cost. That is no compassion. Say the thing plainly and privately, with a path forward.
4. **Praise follows the same rules.** "Great job" is forgettable. Naming the specific behaviour and its effect makes it repeatable, and by contrast makes the manager's corrective feedback credible.
5. **Ask for it, and be seen to act on it.** Upward feedback only arrives where the reaction to the last piece was visibly good. The first time a manager punishes a messenger, the channel closes for everyone.

6. **The truth degrades as it travels upward.** Each layer between a leader and reality filters, softens and summarises. The more senior you are, the more deliberately you have to build channels that carry unfiltered signal, and the more your reaction to bad news decides whether you hear any again.

---

3. Models and evidence

Feedback has more empirical work behind it than most management topics. The practitioner structures in common use rest on practice, though, and have not been tested in trials, so the unit grades them as practice.

Situation, behaviour, impact [practice]

Sloan R. Weitzel, Feedback That Works: How to Build and Deliver Your Message (2000) sets out the structure. Name the specific situation, describe the observable behaviour and state the impact it had. It works because it removes the two things that set off defensiveness, generalisation ("you always...") and diagnosed intent ("you don't care about..."), and because it gives the person something they can change. It gives structure and does nothing for nerve. A difficult message shaped as situation, behaviour and impact and then never delivered is still avoidance. For a pattern, as against a single moment, name the pattern and back it with several specific instances.

Radical candour [practice]

Kim Scott, Radical Candor: Be a Kick-Ass Boss Without Losing Your Humanity (2017) offers a two-axis model (care personally, challenge directly) and describes what goes wrong when either axis is missing. Challenge without care reads as aggression. Care without challenge produces what the book calls ruinous empathy: the well-liked manager whose people never learn where they stand. For this unit its value lies in diagnosing the second failure, which is the common one among managers who think of themselves as kind. The model describes a stance and offers no technique. It explains why to have the conversation; the structure above explains how.

The three triggers [practice]

Douglas Stone and Sheila Heen, Thanks for the Feedback: The Science and Art of Receiving Feedback Well (2014) models why feedback is hard to receive, where most writing looks at why it is hard to give. Truth triggers fire when the feedback seems wrong, relationship triggers when the problem is who it comes from, and identity triggers when it threatens the receiver's sense of who they are. The model helps in both directions. A manager giving feedback can anticipate which trigger a message will set off and shape it to suit. A manager

receiving feedback can notice their own trigger firing and separate the reaction from the content. The book is a practitioner synthesis and is graded as one.

Feedback as a system [practice]

This is a recurring executive practice with no single name: the mechanisms that stand in for the unfiltered signal a leader no longer gets in person. They include skip-level conversations used for signal and kept away from decisions, engagement data that somebody reads and acts on, calibrated reviews of leaders, and norms that make dissent at the leadership table safe. None has a single provenance. All are widely adopted, and all fail in the same way. They exist on paper while the leader's reaction to bad news teaches everyone to stop using them.

---

4. Practice

The weekly small correction

Corrective feedback is given in the week it is earned, in private and in under five minutes. It is shaped as situation, behaviour and impact and followed by a question instead of a lecture. The habit matters more than any single instance. A team that gets small feedback often stops hearing feedback as a verdict, and when the difficult conversation comes it continues something familiar and does not feel like an ambush.

The difficult conversation, prepared

Before a hard conversation the manager writes down three things: the specific instances, the impact on the team or the work, and what they are asking for. If the instances cannot be written down, the conversation is not ready, because the feedback is still an impression and has not yet become an observation. The conversation opens with the behaviour, states the impact, asks for the person's account and closes with a named expectation and a date to look again. Opening with reassurance delays the message, and closing without the expectation leaves the message deniable, so it does neither.

Triangulating second-hand signal

Feedback about someone sometimes arrives through others, especially about a manager from their reports. The practice is to verify before acting, using skip-level conversations, direct observation, peer input and team-health indicators such as attrition and engagement. Taking raw hearsay to the person burns the sources and invites denial. Dismissing it because it has not come through a formal channel abandons the people who provided it.

The feedback eventually given is the manager's own, based on evidence, and does not identify who spoke.

Asking for it

At a fixed interval the manager asks each report one concrete question about their own management, such as what to do more of, what to do less of, or one thing the manager does that gets in the report's way, and then reports back what they changed. Everything depends on the report-back. Asking without being seen to act teaches people that the question is ceremonial.

Protecting the messenger

When bad news arrives, the practice is the visible reaction: thank the person, ask what they need and act. The behaviour is the same whether the messenger is an engineer or a vice-president. Nothing a senior leader does has more effect on the quality of their information.

5. Scaling note

At team scope the object is an engineer's behaviour, and the manager gives feedback directly and often. At organisational scope the object becomes a manager's leadership, which nobody can observe by reading status reports. The manager of managers has to gather evidence at one remove and give feedback about how someone leads as well as what their team ships. At executive scope the truth has largely stopped arriving unprompted. The work then is to institutionalise what lower scopes do in person, and to deliver hard messages to powerful people without flinching and without humiliating them. The pattern is in How Judgment Scales; leading the managers who receive the feedback is PL-6 Leading through leaders.

6. Judgment

- Gives frequent, small, specific corrections so that feedback stops feeling like a verdict. [team]
- Delivers a hard message plainly and respectfully, in private, with a path forward.
- Names the behaviour and its impact, never the person's character or presumed motive.
- Solicits upward feedback and visibly closes the loop on it.
- Praises specifically, so the praise is repeatable and the corrections are credible.
- Failure mode: saving feedback for the performance review. [team]

- Failure mode: softening a hard message until it is unrecognisable, then being surprised the behaviour continued.
- Failure mode: giving feedback while visibly frustrated, so the emotion drowns the content. [team]
- Failure mode: "you're careless", which targets the person instead of the behaviour.
- Gives managers feedback, shaped as situation, behaviour and impact, about how they lead as well as what their team delivered. [org]
- Triangulates second-hand signal before acting on it and protects the people who provided it. [org]
- Names the real gaps of a well-liked manager whose delivery has slipped; likeability is not the bar. [org]
- Failure mode: letting a manager's own account stand in for the reality of their team. [org]
- Failure mode: relaying skip-level input so identifiably that people stop talking. [org]
- Failure mode: treating second-hand signal as either a verdict or noise. [org]
- Builds mechanisms that surface bad news early and rewards the people who bring it. [exec]
- Delivers hard feedback to senior, powerful people with the same directness owed to a junior engineer. [exec]
- Failure mode: softening messages in proportion to the recipient's seniority. [exec]
- Failure mode: punishing messengers and then wondering why surprises keep arriving. [exec]
- Failure mode: mistaking the absence of complaints for organisational health. [exec]

7. Tensions

Directness versus relationship. The fear is that a plain message will damage the working relationship. Usually the reverse holds. The fragile relationship is the one in which hard things cannot be said, and people trust a manager's praise because they know the same manager will criticise. Whether to deliver is settled; the judgment lies in how.

Timeliness versus preparation. Feedback given in the moment is the most useful and the most likely to be badly shaped, while carefully prepared feedback arrives late. Scale resolves it. Small corrections happen in the moment, difficult conversations are prepared, and a difficult conversation never waits beyond the week in which the instances were gathered.

Evidence versus urgency. Second-hand signal about a manager demands action and cannot be acted on as it stands. The harm continues while you verify; acting without verifying destroys sources and may be wrong. The discipline is to verify quickly and deliberately instead of choosing one over the other.

Candour versus safety. A team that often hears hard truths can become a team that braces for them. The counterweight is the care axis. Hard messages land within a relationship in which the manager has visibly invested in the person, praised specifically and asked for feedback in return.

Filtering versus fidelity. As information travels upward some summarising is necessary, and every bit of it costs fidelity. A leader who demands raw detail drowns in it, and one who accepts summaries ends up managed by them. The judgment is in deciding which signals must arrive unfiltered and building the channel for those.

8. Worked scenario

Over two weeks, in three separate skip-level conversations, a manager of managers hears that one of their managers takes credit for the team's work in leadership meetings and is dismissive of people in private. That manager runs a high-output team and is well regarded by peers, and their own account of the team in weekly one-on-ones is uniformly positive. Nothing has reached a formal channel. Two of the three engineers who spoke asked not to be named.

Both reflexes are wrong. Taking the accounts to the manager directly, as received, exposes the engineers who trusted the skip-level, and invites a denial that cannot be answered without naming them. Dismissing the accounts because they are informal, or because the numbers are good, abandons the people who gave them and tells them the channel is decorative.

The judgment is to verify before acting, and to act as soon as verification allows. The manager of managers widens the evidence without exposing anyone. They attend the next two leadership meetings where the manager presents and listen to how the work is attributed. They look at the team's attrition and internal-transfer requests over the past year, ask the manager's peers an open question about working with the team, and bring forward their remaining skip-levels with that team by a week. Within a fortnight things they have seen for themselves confirm the picture. The manager presented as their own a design that the skip-levels attributed to two engineers, and two transfer requests from the team are pending.

The feedback given is then the manager of managers' own. In a private conversation they name two specific instances they witnessed and state the impact: the team's work is invisible to leadership, and its people are trying to leave. They ask for the manager's account

and do not cite the skip-levels. They set an explicit expectation about attribution and about how the manager speaks to their people, with a date to revisit it against fresh evidence. They also tell the manager that this feedback concerns leadership, that it is separate from delivery, and that good delivery does not offset it.

Two things are deliberately left undone. The engineers who spoke hear only "I heard you, and I am acting on it" about what happened to their words, which closes the loop without the detail that would expose them. And the manager's output is not treated as a reason to wait. The version of this story that was avoided is the one in which the manager of managers waits for a formal complaint that never comes, because the people who would have made it have already left.

9. Related competencies

- PL-4 Performance management: where the conversation goes when feedback has not changed the behaviour, with expectations, support and honest consequences.
- PL-6 Leading through leaders: the managers who receive feedback about their leadership, and how their judgment is developed.
- CC-1 Psychological safety and productive conflict: the safety that decides whether feedback, especially upward, is given at all.
- CC-2 Coaching versus directing: coaching as the form feedback takes when the goal is judgment more than a specific behaviour.
- CC-5 Values, integrity, and ethical judgment under pressure: the costly right call when the person who needs the hard message is senior, valuable or feared.

10. Self-check

1. Rewrite "you need to communicate better" as a situation, a behaviour and an impact. *Answer:* "In Tuesday's design review, when the platform team raised the migration risk, you moved on without responding; they left believing the risk was accepted, and it was not." It gives a specific moment, an observable act and a named consequence, and says nothing about the person's character.
2. Why does softening a hard message until it is unrecognisable count as a failure of kindness rather than an excess of it? *Answer:* The person will discover the real message later, from consequences, without having had the chance to change. The manager has spared themselves the discomfort and passed on the cost.
3. Three engineers tell you in skip-levels something damning about their manager. What do you do first, and why not confront the manager immediately? [org] *Answer:* Verify through your own observation, peer input and team-health data, without exposing the

sources. Confronting the manager with hearsay burns the people who trusted you and invites a denial you cannot answer. The feedback you eventually give has to be your own and based on evidence.

4. What is the single most consequential thing a senior leader does for the quality of the information that reaches them? [exec] *Answer:* How they visibly react to bad news. Thanking the messenger and acting keeps the channel open for everyone watching; punishing a messenger once closes it for a long time.
5. Name the three triggers that make feedback hard to receive, and say how knowing them helps the giver. *Answer:* Truth, relationship and identity triggers. The giver can anticipate which one a message will set off and shape it to suit: more evidence for a truth trigger, a different messenger or an explicit repair of the relationship for a relationship trigger, and a framing that separates the behaviour from the person for an identity trigger.
6. A well-liked manager's delivery has slipped for two quarters and their team is happy. What is the feedback, and what is not the bar? [org] *Answer:* The feedback names the specific delivery gaps with evidence and sets a clear expectation and a review date. Likeability is not the bar. A happy team that does not ship is a leadership gap, and avoiding the conversation because the manager is liked is the failure this unit describes.
7. What replaces the unfiltered signal an executive no longer receives in person, and what makes each replacement fail? [exec] *Answer:* Skip-level programmes, engagement data that somebody reads, calibrated leadership reviews and norms for dissent at the leadership table. Each fails in the same way: it exists on paper while the leader's reaction to the bad news it carries teaches people to stop using it.

Sources

- Sloan R. Weitzel, Feedback That Works: How to Build and Deliver Your Message (2000): the situation, behaviour, impact structure.
- Kim Scott, Radical Candor: Be a Kick-Ass Boss Without Losing Your Humanity (2017): care personally, challenge directly; ruinous empathy.
- Douglas Stone and Sheila Heen, Thanks for the Feedback: The Science and Art of Receiving Feedback Well (2014): the three triggers; receiving feedback well.

PL-3 Hiring and onboarding

structured evaluation, calibrated standards, assessing capability rather than tool access

1. Definition and why it matters

Hiring is deciding who joins, against a bar defined before anyone is interviewed, through a process that measures capability instead of impression. Onboarding is the work that turns a good hire into a contributing one, and its absence is the usual reason a good hire fails. The competency is structured, calibrated evaluation in place of gut feel. Few decisions a manager makes carry more weight. A weak engineering hire costs a team a year, a weak manager appointment costs every person on that team, and a weak executive hire can cost an organisation years it does not get back. The stakes grow with scope, and so does the temptation to skip the structure, because senior hiring feels like a judgment about people more than a process to be run. It is both. The examinations test hiring because its failure modes are so consistent: the bar lowered because the seat has been open too long, the single impressive signal that colours everything, the senior hire left to sink, and the interview that measured which tools a candidate was allowed when it should have measured what they can do.

2. Core principles

1. **Define the bar before the first interview.** Decide what the role must be able to do, and at what level, and write it down before any candidate is seen. A bar defined during the process ends up defined by the candidates in it.
2. **Structure beats impression, and it is also fairer.** Ask consistent questions mapped to a rubric and score them independently before any discussion. This is one of the better-evidenced claims in management. Structured interviews predict performance markedly better than unstructured ones, and the same structure makes them less biased.
3. **Assess what a candidate can do, not which tools they were permitted.** What matters is whether the person can reason, design, judge and explain. Whether an assistant was in the room matters only as far as the interview shows what the candidate understood of what it produced.
4. **Hold the bar under pressure.** A bad hire costs far more than an empty seat, and the pressure of the open seat is when the bar is most at risk. If the process is not finding people, change the process or the sourcing and leave the standard alone.
5. **Onboarding is where the hire is activated or wasted.** It needs explicit outcomes for the first quarter, an early real contribution, a named guide, and context written down

instead of held as tribal knowledge. The more senior the hire, the more onboarding becomes integration into norms, relationships and how decisions really get made, and the more often its absence turns out to be the cause of failure.

6. **Intervene early on a hire that is not landing.** Three months of thin results and a sceptical team is a signal to diagnose and support now. Waiting politely for certainty at six months is a mistake, and so is concluding on the spot that the hire was wrong.

---

3. Models and evidence

Hiring is unusual among the competencies in this standard because its central claim has strong empirical support, and the unit grades it to match.

Structured interviews [research]

Frank L. Schmidt and John E. Hunter, The Validity and Utility of Selection Methods in Personnel Psychology: Practical and Theoretical Implications of 85 Years of Research Findings (1998) is a meta-analysis covering eighty-five years of selection research. It finds that structured interviews are among the most valid predictors of job performance available, well ahead of unstructured interviews, and that combining them with a work sample or an ability measure improves prediction further. They work by reducing noise, through the same questions, a defined rubric and independent scoring before discussion. On its own, though, structure does not remove bias. A rubric written by one person encodes that person's assumptions, and the remedy is deliberate design and a panel whose members do not all share them.

Calibration [practice]

Calibration means aligning interviewers, and later managers, on what "meets the bar" means, so the same capability earns the same decision whoever assessed it. It has no single provenance. It is the operational partner of structured interviewing and the mechanism that keeps a hiring bar consistent across teams and over time. Without it, structure yields consistent scores that mean different things to different scorers.

The halo effect [research]

Edward L. Thorndike, A Constant Error in Psychological Ratings (1920) described it and it has been replicated many times since: one salient positive trait colours the assessment of unrelated traits. In hiring it is the impressive credential, the strong first answer or the fluent presentation that lifts every later score. The direct countermeasure is independent scoring on each criterion before discussion. A panel debrief that opens with overall impressions

brings back the effect the structure took out. A related bias, affinity (favouring people like oneself), is the reason panels should not resemble the manager.

The first ninety days [practice]

Michael D. Watkins, The First 90 Days: Critical Success Strategies for New Leaders at All Levels (2003) frames a new role as a transition with work of its own: learning, early wins, and building the relationships the role depends on. For engineers this becomes explicit 30-, 60- and 90-day outcomes with a first shippable task. For senior leaders it is mostly integration. The book's central point holds for both, that a transition left to happen by itself usually goes badly. A plan does not replace a relationship, and no checklist replaces the support behind it.

Hiring as a system [practice]

At organisational scale hiring is a system with a throughput and a quality, and Will Larson, An Elegant Puzzle: Systems of Engineering Management (2019) treats it as one. That covers sourcing, a bar that survives growth, calibration across many interviewers, and the choice between internal promotion and external hire, which becomes a portfolio decision instead of a series of one-offs. Seen as a system, the failure shows up before it is felt. A bar that drifts one interviewer at a time cannot be seen in any single hire and is obvious over a year.

---

4. Practice

The role brief and rubric

Before sourcing begins, a one-page brief states what the role must be able to do, the two or three criteria that matter most, and what "meets the bar" looks like for each. The rubric comes from the brief, never from whatever recent candidates happened to show. Every interviewer scores every criterion independently and in writing before the debrief.

The debrief that starts with scores

The panel meets with scores already submitted. The debrief starts with the criterion on which scores diverge most, ahead of anyone's overall impression, and the most senior person speaks last. Divergence is the signal the process exists to surface. Settling early on one strong opinion lets back in the halo the structure removed.

Capability-first exercises

A technical exercise, where one is used, is designed so the candidate's reasoning shows whatever tools they use. They explain the approach, defend the trade-offs and change the result when a requirement changes. An exercise that can be passed by pasting output measures nothing the role needs.

The first-quarter plan

Every hire has a written plan for the first quarter. It sets outcomes at thirty, sixty and ninety days, a first real contribution small enough to ship in the first fortnight, a named guide who is someone other than the manager, and the documents holding the context the team would otherwise carry in its heads. For a senior hire the plan also names the relationships to build and the decisions to watch before making any.

The three-month check

At three months the manager asks themselves two questions. Is this person doing what the brief said the role must do, and does the team think so? If either answer is uncertain, the conversation with the hire happens that week, naming the specific gap and offering support. It is a diagnosis and a repair, and it delivers no verdict.

5. Scaling note

At team scope the object is an individual hire and their first quarter, and the manager runs the process in person. At organisational scope it becomes the team's hiring bar as a system, calibration across teams so that transfers between them work, and the appointment of managers, which is a hire into a new profession and is onboarded as one. At executive scope the object is the hiring system itself, meaning a bar that survives the organisation doubling in a year, along with the senior and executive appointments. There, choosing between an unproven internal candidate and a proven external one has no formula; the choice must be made consciously and resourced whichever way it goes. The pattern is in How Judgment Scales; who is selected to lead, and why, is PL-6 Leading through leaders.

6. Judgment

- Defines the bar first and writes the rubric from it, not from the candidates. [team]
- Runs structured, calibrated interviews and scores independently before discussion.
- Holds the standard when the seat has been open too long, and changes the sourcing instead. [team]

- Designs onboarding for an early real contribution, a named guide, and written context. [team]
- Assesses what a candidate can reason and explain, whatever tools were in the room.
- Failure mode: lowering the bar because the search is dragging. [team]
- Failure mode: over-indexing on a single impressive signal. [team]
- Failure mode: "onboarding" that is a week of reading with no real task. [team]
- Failure mode: assuming a strong hire needs no support. [team]
- Failure mode: a panel that all resembles the manager, and a rubric nobody has examined for what it assumes.
- Calibrates the hiring bar across several teams so the same capability earns the same decision. [org]
- Treats a manager appointment as a hire into a new profession, and onboards it as one. [org]
- Intervenes early and supportively on an external manager hire that is not landing at three months. [org]
- Failure mode: leaving an external hire to sink or swim, then concluding they "weren't a fit". [org]
- Failure mode: hiring bars that drift apart team by team until transfers fail. [org]
- Makes the internal-versus-external bet for a senior seat consciously and resources whichever is chosen. [exec]
- Treats senior onboarding as where the real risk lies, with norms, sponsorship and early direct feedback before any conclusion. [exec]
- Runs a hiring system whose bar survives growth. [exec]
- Failure mode: defaulting to the "safe" external hire and under-supporting them anyway. [exec]
- Failure mode: concluding a struggling senior hire is not a fit without having integrated them. [exec]

7. Tensions

Speed versus standard. Every week a seat stays open costs the team, and every week of pressure wears down the bar. The answer is never to move the bar. Move everything else instead (sourcing, process latency, which items in the brief are must-haves and which are nice-to-haves) and state openly what the empty seat is costing, so the trade-off can be seen.

Structure versus judgment. Structured interviews predict better, and experienced managers feel that structure suppresses what they know. Both are true. Judgment belongs in designing the rubric and reading the evidence. It does not belong in overriding the scores with an impression, which is the very noise the structure exists to remove.

Capability versus fluency. A candidate who presents well, or who quickly produces a polished artifact with assistance, looks capable. The interview has to tell capability from fluency, and doing so takes time and design effort that pressure argues against.

Internal versus external. The internal candidate knows the organisation and has a visible track record, and has not done the job. The external candidate has done it somewhere else and has to integrate. Neither is safe. The discipline is to choose consciously and then pay for the risk of the choice: coaching and air cover for a promotion, deliberate integration for a hire.

Fit versus sameness. A panel converges fastest on candidates who resemble it and calls the result fit. Part of what passes for fit is real, in the sense of shared standards, and part is affinity. The structure exists to tell them apart, and a panel that never disagrees offers no evidence that it has.

8. Worked scenario

A head of engineering has to fill a vice-president seat leading a third of the organisation, and has two candidates. The internal candidate is a director who has run one strong team of teams for two years, is trusted by their people, and has never operated at the new scope. The external candidate has held the equivalent role at a larger company, interviewed well, and would arrive with no relationships here and a very different working culture behind them. The board's chair has said, informally, that the external hire is "the safe choice".

There is no formula. The first judgment is to reject the idea that either option is safe. The external hire carries integration risk, which is usually underestimated and is usually why such hires fail. The internal promotion carries scope risk, which is visible and for that reason usually given too much weight. The head of engineering writes both risks down in the same terms and asks what each choice would commit the organisation to.

They run the process they would run for any hire, at this scope. The role brief comes first. It names the criteria that matter most for this seat, which here are the ability to hold a strategy steady across several organisations and to develop the directors below. Both candidates are assessed against those criteria with the same structure, by a panel that includes peers the seat will work with. Being known does not exempt the internal candidate from the process, and the external candidate gets no credit for scale they have not yet shown here.

On the criteria that matter most, the evidence favours the internal candidate, and the head of engineering makes the promotion. They then resource the decision as the bet it is. There

is an explicit conversation about what the new scope demands and where the director has not yet operated, a coaching arrangement for the first two quarters, and visible air cover: the head of engineering sends questions to the new vice-president in front of the leadership team and does not answer for them. The board chair hears the reasoning, the criteria and what the organisation is doing to make the bet succeed.

Had the evidence pointed the other way, the same discipline would have applied in reverse. The external hire's first ninety days would have been treated as where the risk lay, with explicit norms, a sponsor on the leadership team, and direct feedback in the first month instead of a verdict in the sixth. In neither case does the head of engineering choose by asking which option will be easier to defend if it fails.

9. Related competencies

- PL-6 Leading through leaders: who is selected to lead, on what evidence, and what they are owed in their first six months.
- CC-3 Inclusive practices and equitable opportunity distribution: affinity, panels that resemble the manager, and the fair distribution of opportunity, which begins at hiring.
- PL-4 Performance management: what happens when a hire is not meeting the bar, with expectations, support and honest consequences.
- PL-5 Career development and growth paths: the growth path a hire is joining, with internal candidates as its product.
- TJ-5 AI-assisted engineering: assessing capability rather than tool access, and what AI assistance changes about the work being hired for.

10. Self-check

1. Why are structured interviews both more predictive and more equitable than unstructured ones? *Answer:* They reduce noise. With the same questions, a defined rubric and independent scoring, the assessment tracks the candidate instead of the interviewer's mood, first impression or affinity. The mechanism that improves prediction also removes much of the room for bias.
2. The seat has been open for four months and the team is exhausted. What changes, and what does not? *Answer:* Sourcing, process latency and the brief's true must-haves can all change, and the cost of the empty seat should be stated openly. The bar stays where it is, since a bad hire costs more than the empty seat ever did.
3. What is the halo effect and what specific step in the process counters it? *Answer:* One salient positive trait colouring the assessment of unrelated ones. It is countered by

independent scoring on each criterion, submitted before the debrief, and by a debrief that starts with the divergent scores instead of overall impressions.

4. An external manager hire is three months in with thin results and a sceptical team. What do you do, and what are the two mistakes? [org] *Answer:* Diagnose specifically what is not working, name the expectations, and support the correction now. The two mistakes are waiting politely for clarity at six months and concluding on the spot that the hire was wrong. Most senior-hire failures are integration failures that nobody addressed.
5. How should an interview establish capability when candidates may use AI assistance? *Answer:* By making reasoning visible whatever the tools. The candidate explains the approach, defends the trade-offs, and changes the result when a requirement changes. An exercise that can be passed by pasting output measures nothing the role needs.
6. When does an unproven internal candidate beat a proven external one for a senior seat, and what does the choice commit you to either way? [exec] *Answer:* When the evidence on the criteria that matter most for the seat favours them; there is no formula. A promotion commits you to coaching and visible air cover for the step up, and an external hire commits you to deliberate integration in the first ninety days. Neither option is safe, and the failure is choosing whichever is easier to defend.
7. A technically strong senior hire from a very different company culture is clashing with their new organisation at two months. What comes before any conclusion? [exec] *Answer:* Integration: explicit norms, a sponsor, and early direct feedback. This is usually an integration problem before it is a capability problem, and capability cannot be judged until integration has been attempted.

Sources

- Frank L. Schmidt and John E. Hunter, The Validity and Utility of Selection Methods in Personnel Psychology: Practical and Theoretical Implications of 85 Years of Research Findings (1998): the validity of structured interviews and other selection methods.
- Edward L. Thorndike, A Constant Error in Psychological Ratings (1920): the halo effect.
- Michael D. Watkins, The First 90 Days: Critical Success Strategies for New Leaders at All Levels (2003): the transition as its own work; early wins and integration.
- Will Larson, An Elegant Puzzle: Systems of Engineering Management (2019): hiring as a system; bars that survive growth.

PL-4 Performance management

expectations, support, honest consequences, and fair process

1. Definition and why it matters

Performance management means setting expectations people can see, supporting them to meet those expectations, and following through with honest consequences when they do not. Fair process is the fourth element, and the one v1.0 left implicit. The person knows the bar, has been told where they stand, has been given a real chance, and can see that the same rules apply to everyone. The competency matters for two reasons. Its signature failure, avoiding the hard conversation, is paid for by the rest of the team, who see the gap and quietly lower their own standards to match. Its opposite failure, a consequence delivered without the expectation or the support, is unfair in a way everyone also sees. The examinations test it at every scope: the engineer whose slow delivery had a real cause, the manager who brings a formal plan for someone who was never told the bar, the leader whose results are excellent and whose methods are hollowing out the organisation.

2. Core principles

1. **People can only meet a bar they can see.** Expectations are explicit, written down where it matters, and revisited when the role or the work changes. A performance conversation should refer to something the person already knew.
2. **Early, private and specific.** A gap is raised in the week it is noticed, one to one, with instances. Waiting for the review cycle turns a problem that could have been corrected into a verdict.
3. **Diagnose before responding: skill, will, or conditions.** Either the person does not know how, or they are not applying themselves, or they were not given what the job needed. Each calls for an entirely different response, and the third is more common than managers like to admit.
4. **A formal plan is a real chance, or it is a lie.** A time-boxed, well-supported improvement period with a clear bar is fair. Paperwork for a decision already taken is unfair, and the team knows the difference.
5. **Follow through.** When the period ends and the bar has not been met, the decision was made when the plan was set. Unwinding it teaches everyone that the bar is negotiable and the plan was theatre.
6.

How results are achieved is part of performance. Output produced by fear, by burning out a team or by breaching the organisation's values is a performance problem, and the results do not offset it.

---

3. Models and evidence

Performance management has plenty of process and little evidence. The models here are practitioner tools with clear provenance. The fairness claim rests on a body of organisational-justice research, which the cited article summarises for managers.

Skill and will [practice]

The diagnostic comes from Max Landsberg, The Tao of Coaching: Boost Your Effectiveness at Work by Inspiring and Developing Those Around You (1996). Before responding to underperformance, separate whether the person cannot (a skill gap, met with teaching and support) from whether they will not (a will gap, met with expectations and consequences). This unit adds a third branch that the original does not name, the conditions gap, where the person was never given what the job required: clear requirements, access, time, a functioning team. Treat a skill gap as a will gap and you demoralise the person. Treat a will gap as a skill gap and you coach endlessly at something coaching cannot fix. Treat a conditions gap as either and you produce an unfair review that the whole team recognises for what it is.

Fair process [practice]

W. Chan Kim and Renée Mauborgne, Fair Process: Managing in the Knowledge Economy (1997) sets out three elements (engagement, explanation, and clarity of expectation) drawn from the authors' research into why people accept decisions that go against them. Applied to performance, the person is heard, the reasoning behind the judgment is explained, and what is expected next leaves no room for doubt. This matters because people accept and learn from a consequence they regard as fairly reached, while the same consequence reached unfairly is resisted and remembered by everyone who watched. The grade here is practice because the management version is a synthesis, even though the justice research beneath it is substantial.

Output and leverage [practice]

In Andrew S. Grove, High Output Management (1983), Grove treats a manager's output as the output of the organisation under them. That is why a manager's performance cannot be read from their own report. It has to be read from sustained team health, sustained delivery and how the results were achieved, over time. The same idea explains why a leader who

hits every number by leading through fear is a performance problem: the present number is being bought with the organisation's future capacity.

Calibration [practice]

Calibration, shared with hiring, is the mechanism that makes ratings mean the same thing across teams. It depends on shared standards, on evidence winning over advocacy, and on a joint review when two managers disagree about the same person. Its purpose is fairness, and it typically decays into horse-trading, where ratings are negotiated instead of evidenced. If the most assertive manager's people are consistently rated highest, the meeting is no longer calibrating anything.

---

4. Practice

Expectations, written

Each person has a short written statement of what the role expects at their level and what the next quarter is meant to produce. It is agreed in a one-on-one and revisited when either changes, and it is the document a later performance conversation points to. If it does not exist, the first act of performance management is to write it, before any gap is raised.

The early conversation

The first performance conversation is small. It covers the gap, an instance or two, the question of what is getting in the way, and an agreed next step. It happens in the week the gap is noticed and nobody calls it a warning. Most performance problems that reach a formal stage got there because this conversation was skipped.

The diagnosis before the plan

Before any formal step, the manager writes one paragraph answering three questions. Were the expectations explicit? Was feedback given and documented? Is this skill, will or conditions? If the answer to either of the first two is no, the formal step is premature and the manager's own gap comes first.

The improvement period

A formal plan states the specific bar, the specific support (pairing, reduced scope, a named reviewer), the duration, and what happens at the end. It is reviewed weekly. At the end, the outcome is the one the plan defined, delivered as directly as the plan was set.

Reading a manager’s performance

For someone who leads others, performance is read over at least two quarters from their team's health (attrition, engagement and what skip-levels reveal), their team's delivery, and how the results were produced. One bad quarter with a cause outside the manager's control is judged on the quality of their decisions and their response, not on the raw outcome. One good quarter produced by heroics is not evidence of much.

5. Scaling note

At team scope the object is an engineer's performance against expectations the manager set, and the manager acts directly. At organisational scope the manager of managers has two jobs. One is quality control on how their managers handle performance, which includes declining a formal plan built on absent expectations and coaching the manager first. The other is judging the managers themselves, on evidence gathered over time instead of their own status reports. At executive scope the object is leaders whose results are tangled up with context, politics and the executive's own public bets. Fairness then means calibrating across organisations by the change a leader produced, setting aside how healthy the organisation they inherited was. The pattern is in How Judgment Scales; the managers whose performance is being read are the subject of PL-6 Leading through leaders.

6. Judgment

- Sets expectations the person can see, and revisits them when the work changes. [team]
- Raises a gap early, privately and with instances, before it becomes a crisis. [team]
- Separates skill from will from conditions, and responds to the one that is in fact present.
- Treats a formal plan as a real, supported chance with a defined end, and follows through at the end.
- Protects the team by making a hard decision about one person when the damage outweighs the output, including the high performer whose behaviour breaches the team's values (see CC-5 Values, integrity, and ethical judgment under pressure).
- Failure mode: hoping the problem resolves itself. [team]
- Failure mode: vague feedback that never names the gap. [team]
- Failure mode: endless coaching of a will problem. [team]
- Failure mode: a formal plan for someone who was never told the bar.
- Declines a manager's formal plan when expectations were never explicit, and coaches the manager to set them first. [org]

- Judges managers on sustained team health, sustained delivery and how results were achieved, over time and in context. [org]
- Runs calibration as a fairness mechanism (evidence over advocacy, joint review when managers disagree) and never as a ranking ritual. [org]
- Names an equity problem when a manager gives all the high-visibility work to favourites; opportunity is part of compensation (see CC-3 Inclusive practices and equitable opportunity distribution). [org]
- Owns a headcount-reduction selection with consistent, defensible criteria after hearing the manager's case in full. [org]
- Failure mode: approving a plan built on absent expectations because "the manager owns their team". [org]
- Failure mode: judging managers by their own status reports or by likeability. [org]
- Failure mode: punishing outcomes without reading context, or excusing patterns because of it. [org]
- Failure mode: letting calibration decay into horse-trading. [org]
- Holds leaders accountable for how results are achieved. A leader who delivers through fear while capable people leave is a performance problem, without qualification. [exec]
- Gives a loyal, long-tenured leader whom the organisation has outgrown honesty, and either a reshaped role or a dignified transition, instead of quiet marginalisation. [exec]
- Acts on the evidence when a leader the executive publicly backed is failing; doubling down to protect the bet loses both the organisation and the reputation. [exec]
- Calibrates leaders across organisations by the change they produced and the quality of their decisions, setting aside the health of the hand they were dealt. [exec]
- Failure mode: tolerating a feared-but-effective leader until the pipeline beneath them is hollow. [exec]
- Failure mode: letting loyalty freeze the organisation chart. [exec]

7. Tensions

Care versus clarity. The manager wants to protect the person from a hard message, and the person needs that message to have any chance of meeting the bar. Clarity delivered with care resolves it. Care offered in place of clarity is the failure this whole unit describes.

Fairness to the individual versus fairness to the team. A long improvement period is fair to the person and costly to the team carrying the gap. The resolution is a period that is properly time-boxed and properly supported, so that the person and the team both know when it ends.

Context versus accountability. Reading a bad quarter in context is fair. Reading every bad quarter in context is how patterns get excused. The judgment lies in the timescale: a single quarter is weighed by the decisions and the response, while a pattern across quarters is itself the performance.

Results versus methods. A leader with excellent numbers and departing people forces the question of what performance is. This standard holds that the methods are part of the result, since what is being spent is the organisation's future capacity. The pull toward the number is strong all the same, and the examination tests whether the candidate resists it.

Loyalty versus scope. Tenure and past contribution earn respect and candour. They do not earn permanent scope. Honouring the first while refusing the second is uncomfortable and is the only fair option. The alternatives are an organisation that cannot grow past its early leaders, or a quiet demotion that nobody was honest about.

8. Worked scenario

A manager who reports to a senior manager arrives with a formal improvement plan for one of their engineers, drafted and ready to issue. The engineer's delivery has been slow for two quarters. Reading the plan and the history, the senior manager finds that the engineer's objectives were never written down, that the feedback over the period amounts to two informal comments in one-on-ones, and that by the manager's own account the requirements for the engineer's main project were unclear until six weeks ago.

Both easy responses are wrong. Approving the plan because the manager owns their team hands an unfair consequence to someone who was never given a fair chance, and everyone on that team will read it correctly. Taking the case over, by rewriting the plan or talking to the engineer directly, settles the case and teaches the manager nothing, while telling their team that the manager is not really in charge of it.

The senior manager declines the plan and explains why to the manager, as feedback on the manager's own performance. Expectations were never explicit, feedback was neither given nor documented, and for most of the period the conditions for the work were poor. The engineer has not yet had the chance the plan claims to give. On the evidence, the engineer's problem is at least partly a conditions gap, and the manager's is a skill gap in performance management.

The senior manager then coaches and leaves the doing to the manager. The manager is asked to write the engineer's expectations for the coming quarter this week and agree them in a one-on-one, to give specific, documented feedback at least every fortnight, and to sort out the requirements problem with the product partner, which is the manager's job whatever happens with the engineer. A date eight weeks out is set to look at the evidence together. If delivery still falls short once expectations are clear and support is real, a formal plan will

then be fair. The senior manager says so now, so that the manager does not take the refusal as protection for the engineer.

The senior manager keeps one thing for themselves, the observation about the manager. A manager who reaches for a formal plan before setting expectations has revealed a gap of their own. The senior manager records it as an item for the manager's development and for their own reading of the manager's performance over the next two quarters.

9. Related competencies

- PL-2 Feedback and difficult conversations: the feedback that comes before any performance conversation, and the difficult conversation itself.
- PL-6 Leading through leaders: managing the performance of managers, and quality control on how they manage performance.
- CC-5 Values, integrity, and ethical judgment under pressure: the high performer whose behaviour breaches the team's values, and the costly right call.
- CC-3 Inclusive practices and equitable opportunity distribution: opportunity as part of compensation, and the equity problem inside performance decisions.
- PL-3 Hiring and onboarding: the hire that is not landing, and the early intervention that comes before any formal step.

10. Self-check

1. Why is skill, will or conditions the first diagnostic question? *Answer:* The responses are entirely different: teaching and support for a skill gap, expectations and consequences for a will gap, and fixing what the person was not given for a conditions gap. Responding before diagnosing produces the wrong one, and the team sees it.
2. What makes a formal improvement plan fair rather than paperwork? *Answer:* A clear bar the person already knew, specific support, a defined duration, a stated outcome at the end, and a real chance of meeting it. A plan written after the decision has been taken is the opposite of fair, and everyone can tell.
3. A manager brings you a formal plan for an engineer who was never given clear expectations. What do you do, and what does it tell you about the manager? [org] *Answer:* Decline the plan, coach the manager to set expectations and give documented feedback first, and set a date to look at the evidence again. It shows the manager has a skill gap in performance management, which becomes an item in their own development and in your reading of their performance.
4.

How do you read a manager's performance, and why not from their status reports? *Answer:* From sustained team health, sustained delivery and how results were achieved, over at least two quarters and in context. Their own reports are the most filtered source there is, and a manager's output is the output of their team.

5. Two of your managers cannot agree on the rating for an engineer who worked across both their teams. What is the resolution, and what are the two wrong ones? [org] *Answer:* A joint review of the evidence against the shared standard. One wrong resolution is seniority, where the more senior manager's view wins. The other is splitting the difference, which rates the negotiation instead of the person.
6. A vice-president hits every number and leads through fear; capable leaders keep leaving beneath them. What exactly is the performance problem, and what does delay cost? [exec] *Answer:* The methods are the problem. The present number is being bought with the organisation's future capacity, and the results do not offset that. Delay costs the pipeline of leaders who leave, and teaches everyone that results buy exemption from how they were achieved.
7. A leader you publicly backed is clearly failing. What is the credibility move? [exec] *Answer:* Act on the evidence. Executives who double down to protect the appointment lose both the organisation and the reputation they were trying to protect. Acting is what credibility looks like here.
8. A long-tenured director whom the organisation has outgrown is loyal to you personally. What are they owed? [exec] *Answer:* Honesty: an explicit conversation, and either a role reshaped around their real strengths or a transition with dignity. Tenure earns respect and candour and does not earn permanent scope. Quiet marginalisation is a form of neglect.

Sources

- Max Landsberg, The Tao of Coaching: Boost Your Effectiveness at Work by Inspiring and Developing Those Around You (1996): the skill and will diagnostic.
- W. Chan Kim and Renée Mauborgne, Fair Process: Managing in the Knowledge Economy (1997): engagement, explanation, clarity of expectation.
- Andrew S. Grove, High Output Management (1983): a manager's output as the output of their organisation.

PL-5 Career development and growth paths

one-on-ones, promotions, the Staff+ track, and the IC/manager choice

1. Definition and why it matters

Career development means understanding where a person wants to go and creating the experiences that get them there, which takes more than conversations. Its instrument is the one-on-one, the recurring, protected meeting that belongs to the report and is kept free of status updates. It is where blockers, frustrations and the earliest signs of disengagement come up, provided someone is listening. Its structure is the growth path: levels and tracks that make "what does the next level look like?" concrete and comparable. That includes the senior individual-contributor track, which lets strong engineers grow without leaving the work, and the honest choice between that track and management. The competency matters because retention is largely decided here, months before anyone resigns. A growth path that exists only in the manager's head gets handed out according to the manager's attention, which is a different thing from handing it out fairly. The examinations test it through the early promotion request, the strong engineer who is bored, the ambitious manager who wants your job, and the succession call with one seat and two ready people.

2. Core principles

1. **The one-on-one belongs to the report.** Status can go through other channels. This meeting is for what does not come up any other way: what is in the way, what they want, what has changed. Cancelling it whenever things get busy signals exactly the wrong priority.
2. **Growth is built from experiences.** Conversations set the direction. What moves a career is stretch assignments, real responsibility and visible work, and a development plan with no experiences in it is a wish.
3. **Make the next level concrete and comparable.** Ladders or competency matrices describe what each level looks like. A promotion then becomes a judgment against a shared description, instead of a measure of how assertive the manager is in the calibration room.
4. **Promotion follows demonstrated judgment, not tenure or desire.** The evidence is that the person already exercises the next level's judgment on real stakes. Promoting someone to keep them, or as a reward for waiting, teaches the wrong lesson to everyone watching.
5.

The senior individual-contributor track is a real path, and management is not a prize. The choice between them turns on the work the person wants to do. A manager's job is to help them make it honestly, which includes describing what management involves day to day.

6. **Watch for disengagement early.** People seldom leave suddenly. Withdrawal, cynicism and a stopped flow of ideas show up first, in one-on-ones, to a manager who is paying attention.

---

3. Models and evidence

Career development is practitioner territory. The models below are widely adopted and clearly attributed. None is a controlled finding, so the unit grades them all as practice.

The one-on-one as the report's meeting [practice]

In Andrew S. Grove, High Output Management (1983), Grove treats the one-on-one as the recurring activity that does the most for a manager's output. Its agenda belongs to the subordinate, and its purpose is for both people to learn about the work and the person. The practical consequences have held for four decades. The meeting is protected and regular, the report sets much of the agenda, and the manager listens more than they talk. The familiar failure has held for just as long: it turns into a status meeting the moment the manager needs one.

Growth ladders and competency matrices [practice]

These are explicit descriptions of what each level looks like for each competency. They make progression concrete and let promotion decisions be compared across people and teams. Camille Fournier, The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change (2017) sets out the engineering-management version, walking through the arc level by level. A ladder describes what is expected and sets no timetable. It decays into collecting checkboxes when levels are treated as tenure milestones instead of demonstrated judgment, and it is only as fair as the calibration that applies it.

The Staff+ track [practice]

This is the senior individual-contributor path (staff, principal and beyond), treated as a career of equal standing with management. Will Larson, Staff Engineer: Leadership Beyond the Management Track (2021) and Tanya Reilly, The Staff Engineer's Path: A Guide for Individual Contributors Navigating Growth and Change (2022) describe it from the inside. Two ideas from that literature matter to managers. The track has several distinct archetypes, so "senior engineer but more so" does not describe it. And the invisible coordinating work

that holds teams together, which Reilly calls glue work, tends to be done by people who then go unpromoted for it. A manager who cannot describe the track cannot offer it honestly.

Readiness [practice]

This is a rule this standard adopts from the literature above and from practice, with no single named model behind it. Someone is ready for a larger scope when they already exercise that scope's judgment on real stakes, before they hold the title. Tenure, ambition and how badly the person wants it do not count as readiness. The rule matters most at succession, where the pull toward whoever has waited longest or asked loudest is strongest.

---

4. Practice

The protected one-on-one

It runs weekly or fortnightly, is never cancelled for a status need, and takes the report's items first. The manager keeps a running note for each person (what they said they wanted, what they said was in the way, what changed) and reads it before each meeting. The note is what makes a pattern visible. Three meetings in a row with no ideas offered is a signal that no single meeting would show.

The growth conversation, quarterly

This is separate from the weekly one-on-one and from the review cycle. It asks where the person wants to be in two years, what the next level would require of them, and which experience in the coming quarter moves them toward it. It produces one concrete assignment, and no plan document.

Answering a promotion request

The manager answers with evidence against the level description: where the person already shows the next level's judgment, where they do not yet, and what would show it. A yes given to keep someone is wrong, and so is a no without reasons. If the answer is not yet, the path is specific enough for the person to watch the gap closing.

Describing management honestly

When an engineer is weighing management, the manager describes the job as it is (the meetings, the ambiguity, the distance from the code, the accountability for other people's work) and asks what the person wants to spend their days doing. The strongest engineer,

having said they do not want to manage, is not pushed into it. A reluctant manager is worse for everyone than an empty seat.

Promotion fairness across teams

Where several teams share a ladder, promotions go through shared criteria, written evidence and a calibration that compares people against the description and keeps their managers out of the comparison. The check is easy. If one team's promotions are consistently faster, either its people are consistently stronger or its manager is consistently more assertive, and the second is more common.

5. Scaling note

At team scope the object is one person's growth and next step, and it happens in the one-on-one. At organisational scope it becomes the growth paths available across several teams and who is getting them: promotion mechanics that are fair across managers, moves between teams, and stretch roles that exist only at department scale. At executive scope the object is the career architecture itself (levels, tracks and the leadership pipeline) and succession as a property of the structure. Every key role, the executive's own included, has a ready or nearly ready successor, and readiness is judged on demonstrated next-level judgment. Developing the managers who report to you is a competency of its own, PL-6 Leading through leaders; the pattern is in How Judgment Scales.

6. Judgment

- Holds one-on-ones consistently, lets the report set much of the agenda, and listens more than they talk. [team]
- Builds growth from real experiences: the stretch assignment, the visible piece of work, the responsibility not yet held. [team]
- Answers a promotion request with evidence, a named gap and a path. [team]
- Describes the senior individual-contributor track and management honestly, and lets the person choose the work they want. [team]
- Recognises withdrawal, cynicism and a stopped flow of ideas as early signals. [team]
- Failure mode: cancelling one-on-ones under pressure. [team]
- Failure mode: hijacking the one-on-one for status. [team]
- Failure mode: discussing growth only at promotion time. [team]
- Failure mode: promoting to retain, or promoting for waiting.
-

Failure mode: drafting a reluctant engineer into management because it is the only visible path to more pay or status.

- Creates growth paths across teams (moves, stretch roles, the senior track) and sees who is and is not getting them. [org]
- Runs promotion fairness across teams with shared criteria, evidence and calibration. [org]
- Decides compensation on value and fairness. Fixes an underpaid engineer's pay because it is wrong, whether or not they threatened to leave, and corrects a package that inverts a tenured colleague's through the compensation process, honestly. [org]
- Failure mode: letting each team invent its own promotion bar, so that each team's ceiling is set by its manager's assertiveness. [org]
- Failure mode: the reactive counter-offer, which teaches everyone that threats work and seldom keeps anyone for long. [org]
- Failure mode: a director who hoards strong engineers and blocks internal moves to protect local output, when talent belongs to the organisation. [org] [exec]
- Designs the career architecture (levels, tracks, the leadership pipeline) and reads it by what it produces. [exec]
- Makes succession an explicit, non-optional expectation of every senior role, including their own. [exec]
- Judges readiness by demonstrated next-level judgment on real stakes, not tenure or ambition. [exec]
- Tells the runner-up for a senior seat the truth about their path, invests in it in earnest, and accepts they may leave. [exec]
- Failure mode: inventing a title to avoid the succession conversation. [exec]
- Failure mode: a pipeline one name deep, and critical initiatives that wobble when one leader is courted by another company. [exec]
- Failure mode: equating readiness with tenure, or with how much someone wants the role. [exec]
- Failure mode: counter-bidding for a leader whose growth was neglected for years. [exec]

7. Tensions

The person's ambition versus the evidence. A talented person wants the next level before they consistently show its judgment. Saying no risks losing them, and saying yes teaches the team that asking works. The way through is a specific path: a no that names exactly what would turn it into a yes, and an assignment that lets them show it.

Retention versus fairness. The counter-offer keeps one person this quarter and distorts equity for everyone who did not threaten to leave. The only version that holds up is fixing real underpayment because it is wrong, before anyone uses it as a bargaining chip.

Local output versus organisational growth. A manager loses their strongest engineer to an internal move and the team's delivery dips. The organisation gains someone who grows and stays. Talent belongs to the organisation, and a manager who blocks the move is optimising for the wrong thing. The dip is still real, and the judgment lies in how the team absorbs it.

Ladders versus judgment. A detailed ladder makes promotion fair and comparable, and it also invites people to tick its boxes. The answer is to treat the ladder as a description of judgment to be demonstrated, never as a list of activities to complete, and to say so in every promotion conversation.

Honesty versus hope. The runner-up for a senior seat, the engineer who will not reach the next level on this team, the manager who wants a role that is not coming: each is owed the truth, and the truth may cost the relationship or the person. Manufactured hope costs more later, and everyone notices it.

8. Worked scenario

An engineer who has been on the team for ten months asks their manager for promotion to senior. They are talented and fast, and have shipped two features that mattered. The manager's honest reading is that the engineer does not yet consistently show senior judgment. They still need scope handed to them, they have not led anything involving another team, and in design reviews they are sharp about their own code and quiet about everyone else's. Twice, the engineer has mentioned a friend at another company who was promoted at the one-year mark.

There are two easy answers. One is to approve the promotion to keep a talented person. The other is to decline with "not yet" and a vague reference to time. The first teaches the team that promotion follows asking and tenure. The second teaches this engineer nothing and leaves them concluding that the manager is the obstacle.

The manager answers with the level description. In the one-on-one they acknowledge the growth in specific terms (the two features, the pace, the quality of the engineer's own work) and then go through the senior description next to the evidence. Senior engineers define their own scope, lead work that crosses a team boundary, and raise the quality of other people's work through review. On the first two there is no evidence yet, because the engineer has not had the opportunity. On the third the evidence points the other way. The manager says plainly that the answer is not yet, and that the reason is the evidence and has nothing to do with time served.

The manager then builds the path out of experiences instead of advice. The engineer gets the lead on an upcoming change that touches a neighbouring team's service, with the manager's support at the boundary and a clear expectation that the engineer defines the scope. They are asked to review two pull requests a week from outside their own area and to put the feedback in the review itself instead of giving it privately. The two agree a date three months out to look at the evidence again, and the manager says what would make the answer yes.

The engineer may still leave for the company where the friend was promoted. The manager has judged that a promotion the engineer cannot yet sustain would cost them more than the risk of their leaving costs the team, and has given them an honest path and the chance to walk it. The manager neither promotes to retain nor declines without giving the engineer something to act on.

9. Related competencies

- PL-1 Delegation, ownership, and decision rights: the stretch assignment as delegation, with scope matched to capability plus a deliberate stretch.
- PL-6 Leading through leaders: developing the managers who report to you, and the leadership pipeline as a system.
- CC-2 Coaching versus directing: coaching as the form growth takes when the goal is judgment.
- CC-4 Motivation, recognition, and sustainable pace: motivation, and the early signals of disengagement the one-on-one exists to catch.
- CC-3 Inclusive practices and equitable opportunity distribution: who gets the stretch work and the visibility, and distributing growth paths fairly.

10. Self-check

1. Why should the one-on-one belong to the report, and where do the earliest attrition signals appear? *Answer:* Status can go through other channels. What cannot (blockers, frustration, disengagement) only comes up in a meeting the report controls. The earliest signals are withdrawal, cynicism and a stopped flow of ideas, which show across several one-on-ones to a manager who keeps notes.
2. A talented engineer asks for promotion before consistently showing the next level's judgment. What is the response, and what are the two failures? *Answer:* Acknowledge the growth, go through the level description against the evidence, name the gap specifically, and build a path from real experiences with a date to look again. The two

failures are promoting to retain and declining without giving the person anything they can act on.

3. What distinguishes a real senior individual-contributor track from "senior engineer but more so"? *Answer:* Distinct archetypes with their own scope and influence, equal standing with management, and recognition for the coordinating and quality-raising work that otherwise goes unrewarded. A manager who cannot describe the track cannot offer it honestly.
4. An engineer holds a competing offer and your manager wants to match it. What should the decision turn on? [org] *Answer:* Value and fairness. If the person is underpaid, fix it because it is wrong. If they are not, a reactive match teaches everyone that threats work and distorts equity for those who did not threaten. Retention bought under duress seldom lasts.
5. One team's promotions are consistently faster than the others'. What are the two explanations, and which is more common? [org] *Answer:* Either its people are consistently stronger or its manager is consistently more assertive in calibration. The second is more common, and shared criteria with written evidence are the check.
6. What is the strongest signal that someone is ready for a larger scope, and what are two signals that are often mistaken for it? [exec] *Answer:* That they already exercise the larger scope's judgment on real stakes. Tenure and ambition are the usual substitutes, and neither is evidence.
7. Two directors are ready for one vice-president seat. What do you owe the one who does not get it? [exec] *Answer:* The truth about their path, real investment in it, and acceptance that they may leave. Inventing a title to avoid the conversation weakens the organisation and deceives nobody.
8. What is the quiet measure that an executive's leadership development is working? [exec] *Answer:* Decisions that used to need the executive are made well without them, and the leaders they developed are developing leaders of their own.

Sources

PEOPLE LEADERSHIP

- Andrew S. Grove, High Output Management (1983): the one-on-one as the report's meeting; managerial leverage.
- Camille Fournier, The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change (2017): the management career arc, level by level; ladders.
- Will Larson, Staff Engineer: Leadership Beyond the Management Track (2021): the senior individual-contributor track and its archetypes.
- Tanya Reilly, The Staff Engineer's Path: A Guide for Individual Contributors Navigating Growth and Change (2022): senior individual-contributor leadership; glue work.

PL-6 Leading through leaders

selecting, developing, and holding accountable the managers who report to you

1. Definition and why it matters

Leading through leaders is the job that starts when the people who report to you have people of their own. It means choosing who leads, developing their judgment, and holding them accountable for their teams' outcomes and health without taking the teams back. It is a different job from leading engineers, not a larger version of it. The instincts that made someone a strong team leader, being close to every person and every problem, are nearly the opposite of what the work now asks. The previous edition of this standard treated it as the "Professional band" of delegation, feedback and performance, and so understated how different it is. In this competency a manager's reach becomes greatest at the same moment as their information becomes poorest. It matters because the quality of an organisation's managers sets how well it can execute, and that quality is built or wasted here. The examinations assess it from EMP-II upward. There is no team-scope form, because leading leaders requires leaders to lead.

2. Core principles

1. **Select managers on evidence of judgment about people, not on seniority, tenure or retention.** The sound reasons to move someone into management are a real wish to lead, demonstrated judgment about people in adjacent conditions, and the team's trust. Treat management as a reward and you get managers who do not want the job.
2. **Managers develop through progressively harder real situations with coaching.** Courses and shadowing add to this and cannot replace it. The main instrument is the judgment debrief: going through the hard call the manager faces or has just made, probing the reasoning, and coaching the thinking instead of handing down a ruling.
3. **Hold them to outcomes and health, and do not take the team back.** A manager is accountable for what their team delivers and for the state the team is in. When either slips, respond with coaching and explicit expectations, and do not quietly absorb their job.
4. **Your information is filtered by design; build for it.** A manager's own account is the most filtered source there is. A leader of leaders learns what is true from skip-levels used for signal, direct observation, peer input and team-health data, and never uses the skip-level to bypass a decision.
5. **Span of control is a function of context, not a number.** How many managers or teams one person can lead depends on the managers' maturity, how stable the work is and

how much coordination each report needs. A span that works in steady state fails during change.

6. **The measure is what happens without you.** The system is working when decisions that used to need you are made well without you, and the managers you developed are developing managers of their own.

---

3. Models and evidence

Less has been written about managing managers than about managing engineers, and most of it is more recent. One large internal study is graded as research below. The rest is practitioner work with clear provenance.

What good managers do [research]

David A. Garvin, How Google Sold Its Engineers on Management (2013) reports Google's multi-year internal study of its own managers. The study set out to test whether managers mattered at all and found, against the grain of the company's engineering culture, that they did. It identified a short list of behaviours that distinguished the best: coaching, empowering the team instead of micromanaging, showing interest in people's wellbeing, being results-oriented, communicating, supporting career development, having a clear vision, and having enough technical skill to help. It is a single-company study and the list holds no surprises. Its value is as evidence that these behaviours can be measured, and that the difference they make is large enough to show up in retention and satisfaction data. It is the nearest thing to research evidence this competency has.

Managing managers [practice]

Camille Fournier, The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change (2017) has a chapter on the move to leading managers and the failures typical of it: skip-level meetings used badly, the temptation to keep doing the old job, and the new manager who needs coaching where a replacement is being considered. This unit adopts Fournier's practical point. The people who report to you now need you to hold them accountable for their teams and to develop their judgment, and both are harder than managing engineers because the feedback loop is longer and the evidence arrives filtered.

Sizing teams and spans [practice]

Will Larson, An Elegant Puzzle: Systems of Engineering Management (2019) takes a systems view of how many people a manager can lead and how many managers a leader of managers can support. It argues that organisational design should follow from those limits, whatever that does to the tidiness of the org chart. The specific numbers depend

on context, and the book says so. What carries over is the habit of treating an overloaded manager as a design problem before treating it as a performance problem.

Task-relevant maturity, applied to management [practice]

This applies Grove's concept from Andrew S. Grove, High Output Management (1983) to the case it fits best. A brilliant engineer newly promoted into management has low task-relevant maturity at management itself, however senior they are technically, and needs close support on exactly the things they have never done. The leader of leaders sets the level of support by the manager's experience with this kind of situation (their first performance conversation, their first hire, their first reorganisation), whatever their title.

Coaching managers through change [practice]

Lara Hogan, Resilient Management (2019) deals in practical terms with how a leader supports managers through the transitions their teams go through, such as growth, reorganisation and loss. It favours coaching questions over answers and sponsorship over protection. What it offers here is the distinction between coaching a manager and rescuing them, which is the tension this whole competency turns on.

---

4. Practice

The manager one-on-one as judgment debrief

The weekly meeting with a manager is no project review. At its centre is the hard call they are facing or have just made: what they saw, what they considered, what they chose and why. The leader of leaders asks questions that expose the reasoning and offers a view of their own only after the manager has run out of theirs. Status goes in writing.

The first six months of a new manager

A new manager is being onboarded into a new profession. That calls for an explicit conversation about what changes, a named peer they can ask the embarrassing questions, protected room to learn, and a leader who coaches them to manage outcomes instead of tasks. A new manager who reviews every pull request and redoes junior work does not lose their duties at the first stumble. They are coached to work out what to delegate and what good enough looks like, and protected while they adjust.

Skip-levels for signal

These are regular one-on-ones with the people who report to your managers. Their purpose is unfiltered signal about team health and about issues a manager might miss, and visible accessibility. They are never a channel for decisions that belong to the manager, and what is said in them is never passed on so identifiably that people stop talking. Anything heard is verified before it is acted on, and the action goes through the manager.

Reading a manager

Once a quarter, for each manager, the leader writes three lines: the team's health (attrition, engagement, what skip-levels show), the team's delivery, and how the results were produced. These are read across several quarters and in context. A manager whose personal heroics hide a team that cannot function without them is a risk, however successful they look.

The overloaded manager

When a manager is still doing a lot of individual work on top of managing and is visibly struggling, the leader helps them shed it before they burn out. The leader reads the situation first as a design question (is the span right, is the team the right shape) and only after that as a performance one.

The inherited leadership team

A leader who inherits managers of mixed quality assesses each of them individually against the bar over the first months, on evidence, and then acts deliberately. A purge of the previous leader's people is wrong, and so is letting things drift indefinitely.

---

5. Scaling note

This competency has no team-scope form. An Associate leads engineers, and the EMA-I outline lists PL-6 as not assessed. At organisational scope the object is the managers who report to you, meaning their selection, development and accountability, and the typical failure is reaching around them to their teams. At executive scope the object becomes the leadership team as a system. It needs complementary strengths, clear scopes and shared accountability, where the alternative is a set of individually strong rivals for your attention. It also covers the leaders of leaders beneath you, whose readiness for larger scope you have to see before they ask, and the decision to replace a senior leader when coaching has not worked. Delegating a team is PL-1 Delegation, ownership, and decision rights; designing the structure the managers sit in is SV-6 Organisational design and leading change; the pattern is in How Judgment Scales.

---

6. Judgment

- Promotes people into management for the right reasons and supports the transition deliberately. [org]
- Selects a lead for a new team on demonstrated judgment in adjacent conditions, not on who is loudest, most senior or most available. [org]
- Understands what a strong engineer who does not want to manage wants instead, and develops or finds another leader. [org]
- Runs manager one-on-ones as judgment debriefs and stretches managers with real situations. [org]
- Coaches the new manager who micromanages toward outcomes, and does not strip their duties. [org]
- Helps an overloaded manager shed individual work before they burn out, and asks whether the span is right. [org]
- Uses skip-levels for signal, verifies what is heard, protects the sources, and acts through the manager. [org]
- Reads a manager's effectiveness from sustained team health, delivery and methods, over time and in context. [org]
- Names a manager's avoidance (the hard conversation not had, the one-on-ones that quietly stopped) as a leadership gap, and coaches it. [org]
- Channels a manager's ambition, including ambition for your own role, into growth and scope relevant to succession, and names corrosive politicking directly as behaviour. [org]
- Failure mode: using management as a seniority prize or a retention bribe. [org]
- Failure mode: drafting a reluctant engineer into leadership. [org]
- Failure mode: running manager one-on-ones as project reviews. [org]
- Failure mode: developing managers by course catalogue. [org]
- Failure mode: reaching around a manager to their engineers because you could fix it faster. [org]
- Failure mode: treating every escalation as yours to solve, so that the manager's job quietly becomes yours. [org]
- Failure mode: taking a team back when it struggles instead of developing its manager. [org]
- Failure mode: tolerating a manager whose personal heroics hide a team that cannot function without them. [org]
- Failure mode: treating an ambitious manager as a rival, or ignoring corrosive politicking because the manager is talented. [org]

- Builds a leadership team as a system with complementary strengths, clear scopes and shared accountability, in place of a set of strong individuals competing for attention. [exec]
- Develops leaders of leaders, and sees the signs that a leader is ready for larger scope before they ask. [exec]
- Holds a senior leader accountable for an organisation's outcomes and health while acting through them, and knows when coaching has not worked and replacement is the call. [exec]
- Gives leaders political context instead of political shelter: enough to operate, without the noise. [exec]
- Assesses an inherited leadership team individually and acts deliberately, with neither a purge nor drift. [exec]
- Failure mode: a leadership team that behaves as rivals for the executive's attention. [exec]
- Failure mode: "protecting" leaders into naivety. [exec]
- Failure mode: a senior leader who refuses to develop successors, tolerated because they are excellent. [exec]
- Failure mode: leadership development as a programme owned by a function, when it should be an executive responsibility. [exec]

7. Tensions

Coaching versus rescuing. A manager who reports to you is drowning, and doing their job would be faster. Each time you do it, they learn that the job is not theirs, and so does their team. The only help that leaves the manager stronger is the slower kind.

Accountability versus protection. A new manager needs room to learn, and their team needs a manager who meets the bar now. Explicit expectations with a time horizon resolve it. Protected room has a limit, and the manager knows when the protection ends.

Signal versus bypass. Skip-levels exist because you cannot see a team through its manager alone, and every skip-level gives the team a chance to route around the manager. Decisions mark the line. Signal flows up and decisions flow back through the manager, and a leader who crosses that line once has weakened the manager for a year.

Consistency versus autonomy. Managers who lead their teams differently end up with teams that work differently, and some of that difference is the point. The judgment lies in deciding which practices must be the same across teams (the performance bar, the promotion mechanics, the values) and which belong to each manager.

Loyalty versus the bar. The manager who was there first, who was loyal to you and who got the team through a hard year is falling short of what the organisation now needs. They are owed respect and candour, and they are not owed permanent scope. When loyalty freezes the structure, everyone under it pays.

8. Worked scenario

A senior manager with four teams has one manager, promoted eight months ago from the team they now lead, who is visibly struggling. They are still doing a lot of individual work and took the hardest tickets in the last two sprints. Their one-on-ones with their own team have become irregular, and two of their engineers have said in skip-levels that they do not know what is expected of them. Delivery has not slipped yet, because the manager is personally covering the gap, and the manager's weekly account of the team is "busy but fine". The senior manager used to lead this team and knows exactly how they would run it.

The pull is to step in: attend the team's planning, hold the one-on-ones the manager is missing, or quietly take back the two projects most at risk. Any of these would rescue the quarter and finish the manager. The team would learn that its manager is not really in charge, the manager would learn that struggling means losing the job piece by piece, and the senior manager would have taken on a fifth team on top of their four.

The judgment is to diagnose before responding, and then to coach with explicit expectations instead of absorbing the work. The diagnosis is a new manager with low task-relevant maturity at management itself, who has retreated into the work they are good at because the work they are new to is frightening, and whom nobody has told what the job now requires. That makes it first a leadership gap in the senior manager, since eight months in, the expectations of the role have never been made explicit.

In the next one-on-one the senior manager names what they have seen (the tickets, the irregular one-on-ones, the two engineers who do not know what is expected) as situation, behaviour and impact, without passing a verdict. They state the expectation plainly. The manager's output is the team's output, and the team needs a manager more than it needs another engineer. Then they coach the thinking. What would the manager have to stop doing to hold one-on-ones every week? What definition of good enough would let them hand back the hard tickets? Which two things does the team most need to hear about expectations? They agree three concrete changes for the next month and a date to look at the evidence together, and the senior manager says what they will do themselves: nothing in the team directly, and a weekly debrief of the hardest call the manager faced.

The senior manager also asks the design question. Four teams under one senior manager, with one new manager among them, may be the wrong span for this year. If the new manager needs more coaching than the senior manager can give, that is a structural fact

to raise upward, and gives no licence to reach around. The thing the senior manager would do best, running the team, is the thing they leave alone.

9. Related competencies

- PL-1 Delegation, ownership, and decision rights: delegating a whole team to a manager, with reaching around as the organisational form of taking work back.
- PL-2 Feedback and difficult conversations: feedback about how someone leads, gathered at one remove.
- PL-4 Performance management: reading a manager's performance, and quality control on how they manage performance.
- PL-5 Career development and growth paths: the leadership pipeline, with readiness and succession as a system.
- CC-2 Coaching versus directing: coaching the coaches, which means developing judgment in someone who develops it in others.
- SV-6 Organisational design and leading change: the structure managers sit in, with span of control as a design question.

10. Self-check

1. What are sound reasons to make someone a manager, and what are the unsound ones that are more common? [org] *Answer:* A real wish to lead, demonstrated judgment about people in adjacent conditions, and the team's trust. The common unsound ones are seniority, tenure and retention. Treating management as a reward produces managers who do not want the job.
2. A manager who reports to you is drowning. What distinguishes coaching them from quietly doing their job? [org] *Answer:* Who is deciding at the end of the week. Coaching leaves the decisions and the team with the manager and develops their judgment through questions and explicit expectations. Doing their job takes the decisions back and leaves them only the title.
3. What is a skip-level for, and what is it never for? [org] *Answer:* It is for unfiltered signal about team health and issues a manager might miss, and for visible accessibility. It is never for decisions that belong to the manager, and what is said is never passed on so identifiably that people stop talking.

4. How do managers actually get better, and what is the role of training? [org] *Answer:* Through progressively harder real situations with coaching, above all the judgment debrief on the hard call they face or have just made. Courses and shadowing add to this and cannot replace it.
5. A newly promoted manager is reviewing every pull request and redoing junior work while their team's growth stalls. What is the intervention, and what is not? [org] *Answer:* Coach them to manage outcomes instead of tasks, help them work out what to delegate and what good enough looks like, and protect them while they adjust. The wrong intervention is stripping their duties or sending them back to individual work at the first stumble.
6. What determines a healthy span of control, and why is there no magic number? [org] *Answer:* The managers' maturity, the stability of the work, and how much coordination each report needs. A span that works in steady state fails during change, so the right number depends on context and is never a constant.
7. What is the difference between a leadership team and a set of strong leaders, and what does the executive do about it? [exec] *Answer:* A leadership team has complementary strengths, clear scopes and shared accountability, while a set of strong leaders competes for the executive's attention. The executive designs the scopes, makes the shared accountability explicit, and refuses to referee rivalries.
8. What is the strongest signal your leadership development is working? [exec] *Answer:* Decisions that used to need you are made well without you, and the leaders you developed are developing leaders of their own.

Sources

- David A. Garvin, How Google Sold Its Engineers on Management (2013): the internal study of what distinguishes effective managers.
- Camille Fournier, The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change (2017): the transition to managing managers and its failures.
- Will Larson, An Elegant Puzzle: Systems of Engineering Management (2019): sizing teams and spans as a design constraint.
- Andrew S. Grove, High Output Management (1983): task-relevant maturity, applied to the new manager.
- Lara Hogan, Resilient Management (2019): coaching managers through change; sponsorship over protection.

Delivery & Execution

What this domain is

Delivery & Execution is the work of turning intent into shipped, reliable software, predictably. Good intentions meet their constraints here: limited time, shifting requirements, a finite number of people, and the plain difficulty of estimating software. The domain carries 20% of every examination: 12 questions on EMA-I, 15 on EMP-II and 18 on EME-III. Its five competencies are listed below with the questions each carries at each level, and the Exam Content Outlines give the proficiency indicators.

The throughline

Each competency in this domain is about managing under uncertainty. Strong managers do not promise a certainty they cannot deliver. They build predictability from sequencing, honest forecasting, visible risk, owned reliability and the lightest structure that works. The examinations consistently reward the option that surfaces a trade-off or a truth over the one that makes this quarter look good. So the cost of a yes is named instead of absorbed, a range is given in place of a guessed date, an incident is restored before anyone understands it, a masked slip is met with ground truth, and a ceremony is pruned instead of defended.

How the domain scales

At team scope you run one team's delivery directly. At organisational scope you run the system between teams: the shared bottleneck, the commitment that depends on four teams, the incident that spans three, the dependency nobody confirmed, the standard at the seam. There the truthfulness of status becomes the thing you manage. At executive scope you build the organisation's execution operating system, meaning the portfolio, the planning cadence, a reliability trend that someone owns, and the information flows that let dozens of teams deliver without you in the room. The general pattern is in How Judgment Scales; each unit's scaling note places its own competency on it.

Reading order

Start with DE-1 Prioritisation under constraint, since everything else is sequenced against it. DE-2 Estimation, planning, and honest commitments follows, because a commitment is a promise about a sequence. DE-4 Managing scope, risk, and dependencies across

workstreams is the companion to both. It covers the dependencies and risks that widen a range and re-order a queue, and it holds the domain's treatment of status that has stopped being true. DE-3 Reliability, incident response, and operational ownership stands somewhat apart and can be read on its own. Read DE-5 Designing how work flows last: it describes the system the other four run inside, and it holds the domain's contested ground on AI-assisted work.

Across domains

Delivery relies on Strategy & Vision to say what the sequence is for, through SV-1 Engineering strategy for the strategy, SV-3 Roadmapping for the roadmap and SV-5 Engineering metrics and measurement for measuring outcomes in place of motion. It relies on Technical Judgment for the quality floor, TJ-3 Code review and quality standards, and for the governance of AI assistance, TJ-5 AI-assisted engineering. The costly right call, when the error budget is inconvenient or someone asks for a forecast to be shaded, is CC-5 Values, integrity, and ethical judgment under pressure. When boundary friction keeps recurring, the structure it points at is SV-6 Organisational design and leading change.

DE-1 Prioritisation under constraint

sequencing, trade-offs, saying no

1. Definition and why it matters

Prioritisation is the discipline of deciding what not to do. Demand always exceeds capacity, so the job is to sequence work so that the most important outcomes happen first, and to say no to the rest clearly and early. Ranking a list is the easy part. The competency is the habit of reasoning explicitly about value and cost of delay instead of reacting to whoever asked loudest or most recently, together with the nerve to show the resulting trade-off to the people who lose by it. Without it an organisation has no priorities at all: everything is started and nothing finished, and a team or a whole organisation stays fully busy while producing little. The examinations test it at every scope for two reasons. The failure is easy to recognise (the yes to everything, the thirty priority-ones, the failing initiative kept alive by its sponsor), and the defensible answer nearly always brings to the surface a trade-off that the comfortable answer quietly absorbs.

2. Core principles

1. **Sequence by value and cost of delay, not by advocacy.** Ask what the organisation loses for each week a piece of work goes unshipped, compared with the alternatives. Who asked, how loudly and how recently are not inputs.
2. **Say no by surfacing the cost of yes.** A strong no is never a flat refusal. It takes the form of trade-off transparency ("we can do this, and that slips. Which do you want?"), so the stakeholder makes an informed choice and the manager has not silently absorbed the cost.
3. **Starting less finishes more.** Work in progress trades against throughput. The more that is started, the slower everything finishes, and a system loaded to full utilisation has no room for the surprise that is always on its way. Looking busy and being effective are different things.
4. **Defend focus against churn.** Reprioritising so often that nothing ships is a failure of prioritisation, and should not be mistaken for responsiveness. A priority that changes every week is not a priority.
5. **Own the trade-off at the level where the conflict lives.** When teams compete for a shared bottleneck, or resist lending people to an initiative, the manager above them decides against a single ranked view and names what slips. Refereeing the dispute is not the same as deciding it.

6. **A portfolio is small enough to finish.** At organisational scale, adding parallel initiatives does not add throughput. It adds the appearance of progress while everything slows. The intervention is fewer concurrent initiatives, explicitly ranked and sequenced so they finish.

---

3. Models and evidence

The economics of prioritisation rest on queueing theory, which is mathematics. Its application to management rests on practitioner work with clear provenance, so the unit grades it as practice.

Cost of delay [practice]

Cost of delay is the value lost for each unit of time a piece of work stays unshipped. Donald G. Reinertsen, The Principles of Product Development Flow: Second Generation Lean Product Development (2009) sets it out for product development as the single number that should drive sequencing. Its strength is that it turns arguments about importance into arguments about economics: work with a high cost of delay and a short duration goes first, however senior its advocate. Most delay costs are estimates. The discipline is to treat them as explicit assumptions that people can argue about, and to avoid the false precision that dresses opinion up as arithmetic.

Work-in-progress limits and flow [practice]

The same source supplies the queueing result that a system loaded to full utilisation has unbounded queue times, and its practical corollary: constrain how much is in flight so the most important things finish sooner. Starting less finishes more. The limits of the model are operational. A work-in-progress limit has to be enforced at intake or it becomes decoration, and the slack it creates has to be defended as capacity instead of being harvested as if it were laziness.

Urgent versus important [practice]

This distinction was popularised in Stephen R. Covey, The 7 Habits of Highly Effective People (1989) and is attributed to a remark of Dwight Eisenhower's. It separates what demands attention now from what matters most, because urgency systematically crowds out importance unless someone deliberately resists it. It is a lens and does not schedule anything. Real work is often both urgent and important, and what the distinction asks of a manager is to fund the important work that makes no noise. Labelling quadrants is beside the point.

Trade-off transparency [practice]

This is a practice with no formal name, and the standard treats it as the core of saying no. State what the request costs in terms of what else moves, and let the person who wants it choose. The refusal becomes a decision the stakeholder owns, the manager stops silently absorbing demand, and the reasoning behind the queue becomes visible. Its provenance is ordinary practice. Its absence is the most common prioritisation failure the examinations present.

Portfolio governance [practice]

Portfolio governance is the organisational form of the same discipline. It means an explicit, ranked and deliberately small set of concurrent initiatives, reviewed on a cadence against evidence, with the ranking used to settle contention for resources by strategy instead of politics. It has no single author. Every organisation that finishes things has converged on it, and "thirty priority-ones" is what its absence looks like.

---

4. Practice

The ranked queue with reasons

The team holds its work in one ordered list. Beside each item near the top sits the reason it is there: the outcome it serves and what delaying it would cost. A new request gets placed in the list at a particular position, and that placement is the conversation with the requester. If nobody can explain the order of a queue, it has been accumulated rather than prioritised.

The cost-of-yes conversation

When a stakeholder asks for something new, the manager answers with what would move: "yes, and this slips to next month. Is that the trade you want?" The stakeholder decides. The manager records the decision where the next stakeholder will see it, so the history of the queue is visible and the trade-offs build up into a record instead of a grievance.

Limiting what is in flight

The team agrees how many things it will have open at once and holds to that number at intake. Once the limit is reached, something finishes before anything new starts. Enforcement matters more than the number itself. A limit that gives way whenever someone senior asks is only a suggestion.

One voice to a shared dependency

DELIVERY & EXECUTION

When several teams depend on the same bottleneck, their manager sets one ranked list across them and represents it once, which ends the internal competition. If each team escalates its own request separately, that is three escalations and no decision.

The portfolio review

At organisational scale this is a regular review, based on evidence, of the initiatives in flight: what each is meant to produce, whether it is producing it, and what the ranking implies when they contend for the same people. Sometimes the outcome is a stop. A clearly failing initiative gets the honest recommendation whoever sponsors it, because sponsorship is no reason to keep spending the organisation's capacity.

---

5. Scaling note

At team scope the object is the team's own queue. The manager sequences it directly and protects one team's focus against noise. At organisational scope the object becomes the competing priorities of several teams and the shared bottleneck they contend for. The characteristic move is a single ranked view, represented with one voice, with work in progress limited across the group and the staffing trade-offs owned by the manager instead of refereed. At executive scope the object is the portfolio: which bets the organisation makes, which it declines and which it stops, with the ranking settling contention for scarce senior people by strategy, whoever happens to be shouting. How Judgment Scales describes the pattern, and SV-1 Engineering strategy covers the strategy the portfolio serves.

---

6. Judgment

- Holds an explicitly reasoned queue and can say why each item near the top is there. [team]
- Surfaces the cost of yes and lets the stakeholder choose, instead of absorbing the request. [team]
- Defends the team's focus against the churn of constant reprioritisation. [team]
- Recognises that a team with ten things in progress and nothing shipped has a prioritisation problem, not a capacity problem.
- Failure mode: loudest-voice prioritisation. [team]
- Failure mode: saying yes to everything and quietly missing everything. [team]
- Failure mode: treating every "urgent" as important. [team]
- Failure mode: reprioritising so often that nothing ships. [team]

- Sets one ranked view of demand across several teams and represents it to shared dependencies with one voice. [org]
- Limits work in progress across the group so that the most important things finish. [org]
- Owns the staffing trade-off for a cross-team initiative (decides against the department's priorities and names what slips) instead of refereeing three managers' reluctance. [org]
- Fixes intake as a system: one visible front door, stakeholders redirected consistently, engineers backed when they redirect. [org]
- Failure mode: letting each team fight its own escalation war. [org]
- Failure mode: measuring health by how busy every team is. [org]
- Failure mode: punishing engineers for accepting work the intake system failed to catch. [org]
- Keeps the portfolio small enough to finish, explicitly ranked and sequenced. [exec]
- Uses the ranking to settle contention for scarce senior people by strategy, once, instead of by sponsor. [exec]
- Recommends stopping or descoping a failing initiative regardless of its sponsorship. [exec]
- Leads a crisis reallocation through explicit priorities and through leaders, over-communicating the reasoning. [exec]
- Failure mode: thirty priority-ones. [exec]
- Failure mode: measuring the organisation by initiatives started. [exec]
- Failure mode: letting sponsorship immunise failing work. [exec]
- Failure mode: personally micromanaging a crisis shuffle instead of leading it. [exec]

7. Tensions

Focus versus responsiveness. A queue that never changes ignores new information, and one that changes every week ships nothing. The judgment lies in how big a change has to be before it justifies re-sequencing, and in making each re-sequence an explicit decision with a stated reason instead of a reflex.

Local versus global. If every team maximises its own throughput, the whole falls out of alignment. If every decision is taken centrally, the parts slow down. The answer depends on where the ranking lives: priorities and boundaries are set at the level that can see the whole, and the teams sequence their own work inside them.

Utilisation versus flow. A fully busy team looks efficient and is one surprise away from missing everything. Deliberate slack looks like waste, yet it is what lets the important things

finish on time. Part of the competency is defending that slack against people who read it as idleness.

Sponsorship versus evidence. A failing initiative with a powerful sponsor is politically expensive to stop and organisationally expensive to continue. The honest recommendation is owed whoever the sponsor is. The judgment is in how it is delivered: with the evidence, with options, and early.

Decisiveness versus consensus. In a crisis, a clean priority call made quickly beats a consensus reached late. In steady state, a ranking imposed without hearing the teams will be resisted and gamed. The scope of the situation decides which mode applies. The failure is running in crisis mode all the time, or seeking consensus in the middle of a crisis.

8. Worked scenario

A head of engineering inherits an organisation of twelve teams with thirty initiatives in flight, each labelled a top priority by someone. Nothing has finished cleanly in two quarters. Every team is fully allocated, several engineers are on three initiatives at once, and two strategic programmes, whose sponsors both sit on the executive team, are fighting over the two most senior engineers in the organisation. The previous head of engineering responded by adding a programme office and a weekly status meeting, which produced a very good dashboard.

The temptation is to fix the coordination with better tracking, clearer dependencies and a more senior programme manager. That treats the symptom. Thirty concurrent initiatives across twelve teams is an absence of prioritisation, and no amount of visibility will make them finish. At this scale, adding parallel initiatives does not add throughput. It adds the appearance of progress while everything slows down.

The head of engineering forces a real ranking. Working with the executive team, they list the thirty initiatives, attach to each the outcome it serves and the cost of delaying it by a quarter, and rank them. Then they draw a line. The organisation will run the top eight to completion, sequence the next ten behind them, and stop or fold the rest. The stopped initiatives are named, with the reason, to their sponsors and to the organisation, because an initiative that is quietly starved does more harm than one that is explicitly stopped. The two senior engineers go to the higher-ranked of the two strategic programmes, and the other sponsor hears the reason from the ranking instead of from a negotiation.

Two further choices are deliberate. The ranking is published, so the next contention for a scarce person is settled by looking at the list without anyone escalating. The programme office stays, but its weekly meeting changes purpose: it stops reporting status and starts examining evidence against the ranking, asking whether the top eight are producing what they were meant to and whether any of them should stop.

The costs are real. Three sponsors are unhappy, one initiative with a powerful backer is among those stopped, and the organisation spends a month re-sequencing. What the head of engineering does not do is keep thirty things alive to avoid that unhappiness, because that choice is what produced two quarters in which nothing finished.

9. Related competencies

- DE-2 Estimation, planning, and honest commitments: the commitments that follow from the sequence, and the honesty about uncertainty they require.
- DE-4 Managing scope, risk, and dependencies across workstreams: the dependencies and shared bottlenecks a ranked view has to account for.
- SV-1 Engineering strategy: the strategy the queue and the portfolio exist to serve.
- SV-3 Roadmapping: roadmapping as the same trade-off between product delivery and platform investment, over a longer horizon.
- PL-1 Delegation, ownership, and decision rights: delegating the sequencing within boundaries the manager sets.

10. Self-check

1. Why is "we can do this, and that slips. Which do you want?" stronger than a flat no? *Answer:* It turns a refusal into a decision the stakeholder owns, makes the trade-off visible instead of absorbing it, and puts the reasoning behind the queue on the record. A flat no invites escalation, whereas a visible cost invites a choice.
2. A team has ten things in progress and has shipped nothing in three weeks. Is the problem capacity? *Answer:* Almost never. Work in progress trades against throughput, and starting less finishes more. The intervention is a limit on what is in flight, enforced at intake. More people will not fix it.
3. Three of your teams each escalate a competing request to the same platform team. What does strong judgment do that forwarding the escalations does not? [org] *Answer:* It sets a single prioritisation across the three teams and represents it to the platform team once, which ends the internal competition. Forwarding three escalations amounts to no decision at all.
4. Three managers each refuse to lend people to a cross-team initiative. Whose trade-off is it? [org] *Answer:* Yours. Decide against the department's priorities, name what slips as a result, and own that cost. If you referee the reluctance, the initiative starves politely.
5. Why does adding parallel initiatives reduce an organisation's throughput, and what does that imply for portfolio size? [exec] *Answer:* People and shared dependencies

are then contended for across more work. As utilisation approaches full, queue times grow without bound, and everything slows while the appearance of progress rises. The portfolio should be small enough to finish: fewer concurrent initiatives, explicitly ranked and sequenced.

6. An initiative is clearly failing and has a powerful executive sponsor. What is owed? [exec] *Answer:* The honest recommendation (stop or descope), with the evidence and options, early. Sponsorship is no reason to keep spending the organisation's capacity. Whether to deliver the recommendation is not in question; the judgment is in how it is delivered.
7. What is the difference between leading a crisis reallocation and micromanaging one? [exec] *Answer:* Leading means making the priority call cleanly, delegating the execution of the shuffle to leaders, and over-communicating the reasoning. Micromanaging means personally moving people, which stops the executive from running the system that runs everything else.

Sources

- Donald G. Reinertsen, The Principles of Product Development Flow: Second Generation Lean Product Development (2009): cost of delay; work in progress, utilisation and queue times.
- Stephen R. Covey, The 7 Habits of Highly Effective People (1989): the urgent-versus-important distinction as popularly framed.

DE-2 Estimation, planning, and honest commitments

managing uncertainty rather than chasing accuracy

1. Definition and why it matters

Estimates are unreliable by nature. At the start of a piece of work you know the least you ever will, the happy path is the easiest one to imagine, and the hidden work often outweighs the visible. So the competency is not producing accurate estimates. It is managing the uncertainty honestly, which means estimating in ranges with named assumptions, slicing work small so errors compound less, de-risking before committing, and keeping the line between an estimate and a commitment sharp enough that everyone knows which one they are hearing. Delivery credibility is built or destroyed here. When an optimistic estimate posing as a commitment is defended until the deadline and then missed, the cost goes beyond the slip. It also spends the trust that would have made the next honest forecast believable. The examinations test this because the pressure to give a confident date never lets up at any scope, and the defensible answer is nearly always the one that names what is not yet known.

2. Core principles

1. **An estimate is not a commitment, and both are said aloud.** "Our best estimate is this, and what we can commit to is that." The gap between the two is the buffer against uncertainty. Naming it, where it could have been hidden, is what builds trust.
2. **Ranges, not points.** A single date at the start of uncertain work is a range in disguise. State the range and the assumptions that set its width.
3. **Slice small and de-risk first.** Small pieces are estimated more accurately and their errors compound less. A time-boxed spike on the fuzziest part narrows the range faster than any amount of planning.
4. **History beats hope.** How long similar work took in the past is a better forecast than how long anyone hopes this piece will take.
5. **Re-forecast the moment reality diverges, and say so.** Surprises sprung at the deadline are how trust dies. The honest form of a slip is early disclosure with options on date, scope and resourcing.
6. **Do not guess under pressure; be decisive about the process instead.** If you are asked for a date on the spot with no estimate, commit to the date by which a credible estimate

will exist, name the key unknowns, and commit to scope. Being decisive about how the answer will be produced is different from inventing the answer.

3. Models and evidence

Estimation is one of the better-studied problems in both software engineering and cognitive psychology, and this unit has one research-grade model alongside its practitioner ones.

The planning fallacy [research]

Daniel Kahneman and Amos Tversky, Intuitive Prediction: Biases and Corrective Procedures (1979) found that people predicting how long a task will take systematically underestimate it, even when they know similar tasks overran in the past. They reason from the specifics of the case in front of them and ignore the distribution of similar cases. The corrective the authors propose is the outside view, which forecasts from the reference class of comparable work, and it is the psychological basis for principle 4. The effect is well established and has been replicated across domains. It is why asking people to try harder does not fix estimation.

The cone of uncertainty [practice]

The cone of uncertainty, given its standard software form in Steve McConnell, Software Estimation: Demystifying the Black Art (2006), is the observation that estimates are least accurate at the start of work and narrow as reality is discovered. The working practices of this unit follow from it: estimate in ranges, slice work small, de-risk the fuzziest parts first, and re-forecast as the cone narrows. It narrows only if someone does the discovering. Uncertainty nobody examines stays wide, and the model offers no excuse for failing to re-forecast once more is known.

Cycle time and historical forecasting [practice]

This practice bases forecasts on measured data about how long work takes, in place of how long people estimate it will take. It is the operational form of the outside view. History predicts only work that resembles the past, so novel work still needs de-risking spikes. A cycle-time distribution is itself a range, and quoting its average as a date reinvents the original error.

Brooks's law [practice]

Frederick P. Brooks Jr., The Mythical Man-Month: Essays on Software Engineering (1975) observed that adding people to a late software project makes it later, because the existing people have to train the new ones and the number of communication paths multiplies. This

is why "add people" is rarely the honest option when a commitment is at risk, and why the options left are date, scope and, occasionally, resourcing applied early instead of late. The law's scope is narrower than its fame suggests. It applies to work that cannot be partitioned, and the judgment lies in telling that work apart from work that can.

Batch size and predictability [practice]

Donald G. Reinertsen, The Principles of Product Development Flow: Second Generation Lean Product Development (2009) argues that predictability is a property of the delivery system rather than of effort. Smaller batches and limited work in progress reduce variance in delivery times more reliably than pressure does. So a team producing wildly inconsistent estimates gets engineering help (smaller slices, historical data, forecast reviews) and no exhortation.

---

4. Practice

The estimate with its assumptions

Any estimate longer than a few days is given as a range, with the two or three assumptions that set its width written beside it. When an assumption turns out to be wrong, the range moves and the reason is plain to see. An estimate whose assumptions nobody can state is an impression.

The commitment, buffered and explained

The outward promise is derived from the estimate with a deliberate, stated buffer, and the stakeholder hears both numbers. The commitment also states the scope it covers, the known risks, and how variance will be communicated. A commitment is a shared understanding of those four things. It is no promise to hit a date whatever happens.

The spike before the plan

Before committing to work with a wide range, the team runs a time-boxed investigation of the fuzziest part, with a defined question and a fixed end. It produces a narrower range, or the discovery that the work is something other than it seemed. A week spent learning that a quarter's estimate was out by a factor of two is the best week of the quarter.

Forecast reviews

At a fixed cadence, the team compares its forecasts with its actuals and asks why they differed. Chronic over-commitment and chronic sandbagging are handled the same way,

because neither kind of forecast carries any information. Both are recalibrated against historical data. A team that delivers exactly to plan every quarter, with no misses, is more likely to be sandbagging than excelling.

The early slip conversation

In the week a commitment becomes doubtful, the stakeholder hears about it, along with the cause and the options: move the date, reduce the scope, or add resourcing where it will help. The conversation happens while there are still choices to make. A slip disclosed late turns a delivery problem into a trust problem.

5. Scaling note

At team scope the object is the team's own forecast and commitment. The manager states ranges, buffers deliberately and re-forecasts in the open. At organisational scope the object becomes commitments that depend on several teams. There the uncertainty has to be aggregated honestly (adding up best cases will not do), and the truthfulness of rolled-up status becomes the thing to manage. Calibrating forecasts across teams is a system to run, and a speech will not achieve it. At executive scope the object is the organisation's planning cadence and the credibility of its commitments to the business and the board. That means staged commitments with checkpoints, credibility repaired through a sequence of smaller verifiable deliveries in place of bolder promises, and the flagship call between date and quality made on total cost over time. How Judgment Scales sets out the pattern, and DE-4 Managing scope, risk, and dependencies across workstreams covers the reporting system that carries status upward.

6. Judgment

- Gives ranges with named assumptions and separates the estimate from the commitment. [team]
- Buffers the commitment deliberately and says so. [team]
- Slices work small and de-risks the fuzziest part with a time-boxed spike before committing. [team]
- Re-forecasts early and openly when reality diverges, with options. [team]
- Declines to give a date on the spot with no estimate, and says what it would take to give one.
- Failure mode: treating an estimate as a promise. [team]
- Failure mode: silent padding, so that nobody knows what the number means. [team]

- Failure mode: committing to the optimistic number to dodge a hard conversation. [team]
- Failure mode: going quiet as the timeline slips. [team]
- Defines a department commitment as scope, known risks and a variance protocol, and guards that definition under pressure. [org]
- Aggregates uncertainty across several teams honestly instead of adding up best cases. [org]
- Treats chronic over-commitment and chronic sandbagging as the same calibration failure and recalibrates both against history. [org]
- Improves predictability through batch size and work-in-progress limits instead of pressure. [org]
- Failure mode: letting a commitment mean "guaranteed date". [org]
- Failure mode: guessing under executive pressure and calling it decisiveness. [org]
- Failure mode: praising the sandbagger's streak. [org]
- Failure mode: demanding better estimates without changing how work is sliced or measured. [org]
- Gives the board a staged commitment: near-term milestones firm, later phases as ranges that tighten at named checkpoints, unknowns stated plainly. [exec]
- Rebuilds credibility after repeated misses by diagnosing the systemic cause and delivering smaller, verifiable commitments in sequence, without promising bigger. [exec]
- Discloses a major miss early with options, while choices still exist. [exec]
- Makes the flagship date-versus-quality call on total cost over time, decides explicitly, and communicates the reasoning. [exec]
- Reads execution health as whether the organisation reliably converts commitments into delivered outcomes, and not as activity. [exec]
- Failure mode: giving the board false precision because it asked for a date. [exec]
- Failure mode: re-promising bigger after each miss. [exec]
- Failure mode: sitting on a known miss until it is undeniable. [exec]
- Failure mode: holding a date by shipping something that will fail at scale. [exec]

7. Tensions

Honesty versus confidence. Stakeholders want a date and read a range as evasion. A point estimate sounds confident and is usually wrong. The way through is to be confident about

the process (the checkpoints, the spikes, the date by which the range will narrow) while staying honest about the number.

Buffer versus credibility. A commitment with too little buffer gets missed. One with too much looks like sandbagging, and if it is always hit, that is what it is. Naming the buffer resolves this, because the stakeholder sees both numbers and the reason for the gap.

Slicing versus coherence. Small pieces estimate well but can lose sight of the whole. Large pieces keep the shape of the work and estimate badly. The judgment is to slice for estimation while planning for the outcome, and to keep the connection between the two views visible.

Early disclosure versus premature alarm. A slip disclosed as soon as it becomes possible spreads alarm about risks that may never materialise, and one disclosed only once it is certain comes too late for options. The rule is to disclose when the probability passes the point at which the stakeholder would want to act, and to say plainly that it is a probability.

Date versus quality. Holding a launch date by cutting quality and slipping the date to protect quality are both real options with real costs. The call is made on total cost over time, whatever this quarter's optics. The failure the examinations present most often is the launch that ships on its date and then fails at scale.

---

8. Worked scenario

A senior manager who leads four teams is in a leadership meeting when a vice-president asks them, on the spot, to commit the group to a launch date. The initiative was scoped in a slide deck a week ago and no team has estimated it. The vice-president wants a date for a customer conversation that afternoon and says, in front of the room, that a good leader should be able to give one.

There are two easy answers, a guess and a refusal. A guess delivered with confidence becomes the commitment the moment it leaves the room, and it will be defended against the evidence for the next quarter. A refusal ("we can't commit without estimating") is true, but it sounds unhelpful and leaves the vice-president with nothing to tell the customer.

The senior manager is decisive about the process and leaves the number open. They explain that the initiative has not been estimated, so a date given now would be a guess passed off as a plan. They promise a credible estimate, as a range with its drivers, by the end of the following week, and they commit today to the scope of the first phase, which is understood well enough to start. They name the two unknowns that will set the width of the range (a dependency on a platform team, and a data migration nobody has sized) and say what the group will do in the coming week to narrow both. For the customer conversation they give the vice-president a sentence that is true: the first phase is committed and starts now, and the full timeline will be set within two weeks.

There is a plan behind the answer. The four teams run a one-week spike on the migration and hold a scoping conversation with the platform team. The estimate that comes back is a range with three assumptions attached. The commitment offered to the vice-president is buffered against that range, states the scope it covers and the risks, and says how variance will be communicated. The vice-president gets a date within two weeks, and it is a date that carries information.

The senior manager does not let the pressure in the room push them into inventing a number. The honest answer costs a moment of discomfort in a meeting. The other answer costs a quarter spent defending a date nobody believed, followed by a launch that either slips or ships broken.

9. Related competencies

- DE-1 Prioritisation under constraint: the sequencing a commitment is made against, and the trade-offs that move when it slips.
- DE-4 Managing scope, risk, and dependencies across workstreams: the dependencies and risks that widen a range, and the status reporting that carries a slip upward.
- SV-2 Communicating and partnering across functions: explaining a range and its drivers to a stakeholder who wants a date.
- CC-5 Values, integrity, and ethical judgment under pressure: declining to shade a forecast under pressure, and absorbing the cost of declining.
- DE-5 Designing how work flows: batch size and flow as the system properties that make forecasts believable.

10. Self-check

1. What is the difference between an estimate and a commitment, and why say both aloud? *Answer:* The estimate is the best forecast. The commitment is what is promised

outward, with a deliberate buffer. Saying both names the buffer, so nobody mistakes one for the other, and that visible gap is what builds trust.

2. Why is a single date at the start of uncertain work "a range in disguise"? *Answer:* At the start you know the least you ever will, so the range is at its widest. A point estimate hides that width without removing it. State the range and the assumptions that drive it.
3. A vice-president asks you on the spot to commit your group to a date for unestimated work. What is the credible response? [org] *Answer:* Commit to a date by which a credible estimate will exist, name the key unknowns and what will narrow them, and commit to the scope that is understood. Be decisive about the process and honest about the number.
4. One team consistently over-commits and misses; another quietly sandbags and always hits. Why are these the same problem, and what fixes both? [org] *Answer:* Both produce forecasts that carry no information. Both are fixed by recalibrating against historical data, using smaller slices, cycle-time distributions and forecast reviews. A speech for one team and congratulations for the other fix neither.
5. Why does adding people to late work usually make it later, and what does that leave as options? *Answer:* The existing people have to train the new ones and communication paths multiply, so the work slows down before it speeds up. What remains is date, scope, or resourcing applied early enough to help, disclosed while there are still choices.
6. What makes a staged commitment more credible to a board than a single date, and what does repeated re-promising cost? [exec] *Answer:* Near-term milestones are committed firmly, later phases are given as ranges that tighten at named checkpoints, and unknowns are stated plainly. That is the only kind of forecast a sophisticated audience should believe. Re-promising bigger after each miss spends the credibility that would have made the next honest forecast believable.
7. The flagship launch can hold its date by cutting quality or slip it. What is the basis for the call? [exec] *Answer:* Total cost over time, whatever this quarter's optics. A launch that fails at scale costs more than a slip. Decide explicitly and communicate the reasoning.

Sources

- Daniel Kahneman and Amos Tversky, Intuitive Prediction: Biases and Corrective Procedures (1979): the planning fallacy and the outside view.
- Steve McConnell, Software Estimation: Demystifying the Black Art (2006): the cone of uncertainty; ranges and slicing.
- Frederick P. Brooks Jr., The Mythical Man-Month: Essays on Software Engineering (1975): adding people to late work.

DELIVERY & EXECUTION

- Donald G. Reinertsen, The Principles of Product Development Flow: Second Generation Lean Product Development (2009): batch size and variance; predictability as a system property.

DE-3 Reliability, incident response, and operational ownership

on-call, SLOs, postmortems, learning from failure

1. Definition and why it matters

Reliability is the property that lets users depend on a system. Incident response is what happens when, briefly, they cannot. Operational ownership is the arrangement under which the people who build a system are the people who live with it. Version 1.0 named only the middle one of these. The competency covers all of them: the on-call system, the service-level objectives that say how reliable is reliable enough, the incident in progress with its inverted priorities, and the review afterwards, which either changes the system or assigns blame. Reliability matters because it is where engineering's promises to the business are tested in public, and because the way an organisation treats failure decides whether it learns anything from it. The examinations test it because the failures are so easy to read: the engineer root-causing while customers are down, the review that hunts for a culprit, the fragile system nobody owns, and the leader who obeys the error budget until it becomes inconvenient.

2. Core principles

1. **When systems fail, priorities invert: restore, communicate, then diagnose.** Mitigation comes before root cause. The cause can wait and the customers cannot.
2. **Someone commands, so everyone else can fix.** A defined incident-command role, decided before the incident, owns coordination and communication. A commander who starts debugging has left the role empty.
3. **Incidents are failures of the system, not of people.** A blameless review asks how the system allowed the error and what to change, and without it problems do not surface at all. Blamelessness does not remove accountability. It is what makes real accountability possible, meaning an honest fix to the system.
4. **A review that changes nothing is a ritual.** Postmortem actions are tracked until they are done, and a review is measured by the recurrences it prevents. The documents it produces do not count.
5. **The team that ships it runs it.** Operational ownership lines incentives up behind reliability, whereas a team that throws code over a wall has no reason to make it reliable. Orphaned systems do not heal, and every incident in one is an ownership decision someone has put off.

6. **Reliability is a stated objective and a spendable budget.** A service-level objective says how reliable is enough. The error budget turns the argument between reliability and velocity into a rule agreed in advance, and the rule is only as good as leadership's willingness to obey it when that is inconvenient.

---

3. Models and evidence

Reliability engineering has a well-documented practitioner literature and one substantial body of research, and the unit grades its models accordingly.

Incident command [practice]

One person owns coordination and communication during an incident so that the responders can concentrate on the fix, and the priorities while it lasts are to restore service, communicate and diagnose, in that order. The structure is adapted from emergency-services practice, and Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy (editors), Site Reliability Engineering: How Google Runs Production Systems (2016) sets it out for software. It is decided before the incident, because once one is under way it is too late. At organisational scale, a cross-team incident needs a commander with authority across all the teams involved. Incident command coordinates people. It is no technical rescue.

Blameless postmortems [practice]

A blameless postmortem asks how the system allowed the failure and what should change in process, tooling or guardrails, and it does not ask whom to blame. Its influential statement is John Allspaw, Blameless PostMortems and a Just Culture (2012), and its operational form is in Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy (editors), Site Reliability Engineering: How Google Runs Production Systems (2016). The mechanism is that honesty about failure becomes safe, which makes reliability something the team can improve. Blameless does not mean free of consequences: patterns of recklessness are performance matters, handled separately through PL-4 Performance management. A review whose actions die in a backlog has produced a document and fixed nothing.

Service-level objectives and error budgets [practice]

Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy (editors), Site Reliability Engineering: How Google Runs Production Systems (2016) proposes an explicit reliability target, with the tolerable shortfall against it (the error budget) treated as a spendable resource that gates risk. With budget left, the team ships. With the budget exhausted, it stabilises. This turns a recurring argument into a shared rule agreed in advance. The limit

of the model is entirely cultural. If leaders ship through an exhausted budget in front of the whole organisation, everyone learns what the real policy is, which is why the model comes up again in CC-5 Values, integrity, and ethical judgment under pressure.

DORA stability metrics [research]

The research programme reported in Nicole Forsgren, Jez Humble and Gene Kim, Accelerate: The Science of Lean Software and DevOps (2018) ran large surveys over several years and found that the organisations with the highest delivery throughput also had the highest stability. Change failure rate and time to restore service improve together with deployment frequency, and do not pull against it. This finding matters to the unit in two ways. At the level of the delivery system, reliability and speed are not traded against each other, and time to restore improves precisely where it is safe to be honest about failure. The metrics measure the system at team level. They are never individual measures and never a leaderboard.

---

4. Practice

The incident runbook, decided in calm

Before any incident, the team settles who commands, how the role is handed over, where communication happens, what the first three actions are, and what "restored" means for each critical service. A team that has to settle these during an outage settles them badly and late.

The first fifteen minutes

Declare the incident, name the commander, mitigate (roll back, fail over, shed load) and post a first customer-facing update, all before anyone looks for the cause. The engineer who wrote the change wants to find the bug, and the commander's job is to stop them until service is back.

The postmortem, run as designed

The review happens within days, and weeks is too late. It is blameless in fact as well as in name. It concentrates on the timeline, the contributing conditions, and what would have caught the problem earlier, and it ends in actions with owners and dates. Those actions go wherever the team's real work goes and are reviewed until they are closed. A recurrence means the review itself needs reviewing.

The ownership map

Every production system has a named owning team, recorded where anyone can find it, with on-call and operational responsibility attached. After any reorganisation the map is the first thing to check, because the systems that fall between the new boundaries are where the next incident will happen.

Reading on-call load

On-call is a workload and gets measured: pages per shift, hours lost from sleep, time spent on toil. When a team's on-call load is rising, its reliability debt is compounding. The manager reads that signal at team scope, and at organisational scope it becomes a trend that someone owns.

5. Scaling note

At team scope the object is an incident in progress and the operational health of one service. The manager runs the incident, defends the blameless review, and treats reliability as a deliverable instead of an interruption. At organisational scope the object becomes reliability across services, the ownership gaps a reorganisation left behind, and the on-call system as a whole. The characteristic moves are to install cross-team incident command instead of joining the debugging, to hold the blameless norm against a stakeholder who wants names, and to close the ownership gaps that make orphaned systems fail. At executive scope reliability is a strategic property, covering investment, risk appetite and what the business is implicitly assuming, with the trend owned at organisational level. The part only the executive can play is what they are seen to do after the most public failure. How Judgment Scales sets out the pattern, and SV-6 Organisational design and leading change covers the structural causes of chronic ownership gaps.

6. Judgment

- Restores service, communicates, then diagnoses, and stops the author of the change from root-causing while customers are down. [team]
- Establishes who is commanding before the incident, and gets out of the way of the fix during it. [team]
- Runs the postmortem blamelessly and tracks its actions to completion. [team]
- Treats reliability as a first-class deliverable the team owns, and not as an interruption to real work. [team]
- Failure mode: root-causing during the outage. [team]

- Failure mode: the "who broke it?" review. [team]
- Failure mode: treating recurring incidents as bad luck. [team]
- Failure mode: building teams that throw code over a wall. [team]
- Failure mode: a postmortem measured by the document it produced.
- Installs a single incident commander with cross-team authority for a cross-team incident, instead of becoming the de facto commander for every incident. [org]
- Holds the blameless review against a senior stakeholder who demands names, and explains that blamelessness is what makes real accountability possible. [org]
- Assigns explicit ownership and funds stabilisation for a fragile area a reorganisation left orphaned. [org]
- Designs an on-call system across several teams that is sustainable, and reads reliability as a trend across services. [org]
- Failure mode: letting a powerful stakeholder turn a review into a trial. [org]
- Failure mode: leaving fragile systems orphaned because reassignment is awkward. [org]
- Failure mode: treating operational load as each team's private problem. [org]
- Makes the organisation's reliability trend visible and owned, with investment behind it instead of exhortation. [exec]
- Argues reliability at the executive table as a strategic property: investment, risk appetite, the resilience the business is assuming. [exec]
- Models blamelessness after the most public failure, including owning the executive-level decisions that contributed. [exec]
- Obeys the error budget when it is inconvenient, because the whole organisation is watching what the real policy is. [exec]
- Failure mode: discovering reliability decay only when customers do. [exec]
- Failure mode: letting the worst incident become the moment blamelessness died. [exec]
- Failure mode: funding features while reliability debt compounds. [exec]

7. Tensions

Restoring versus understanding. Every engineer in the incident wants to know why, and the customers want the service back. The order is fixed, restore first, and the difficulty is that restoring sometimes destroys the evidence of the cause. The commander decides what to capture before the rollback, and has minutes to decide it.

Blamelessness versus accountability. A review that names nobody can look like one that holds nobody responsible. The answer is to keep the review, which is about the system, apart from performance management, which deals with patterns of behaviour and happens separately. Mixing the two ruins both.

Reliability versus velocity. In a single decision the two compete. Across the delivery system the evidence says they improve together. The error budget stops the daily argument from being fought again each day, and it works only if the people who set it obey it.

Ownership versus load. Owning what you ship lines up incentives, and it can also bury a team under on-call for systems it inherited. The judgment lies in reading on-call as a measured workload and treating a rising load as debt to fund, not as the team's private problem.

Structure versus heroics. The manager who knows the system best is the one most tempted to command every incident, and each incident they command is one the team does not learn to run. The structure exists to make the manager unnecessary. A manager who resists it has turned themselves into the single point of failure.

8. Worked scenario

A payment service has been down for eight minutes and checkout is failing for every customer. The on-call engineer, who deployed a change forty minutes ago, has the logs open and is reading stack traces. Three other engineers have joined the channel, each proposing a theory. The manager arrives. Nobody has posted a customer-facing update or said who is coordinating, and the product lead has just asked in the channel what happened.

The manager knows the system and has a theory of their own, so the temptation is to join the diagnosis. They do three other things in the first two minutes. They name the incident commander, themselves for now because nobody else has been designated, and say so in the channel. They ask the on-call engineer a single question, whether the change deployed forty minutes ago can be rolled back, and when the answer is yes they say to roll it back now. And they post a first customer-facing update saying what is affected and that a fix is in progress, with no cause given.

The on-call engineer objects that they are close to finding the bug and that a rollback will lose the evidence. The manager holds the order: restore first. They ask the engineer to capture what they need, a snapshot of the logs and the failing request, in the next sixty seconds and then roll back. Service returns at minute twelve and the customer update is revised. Only then does the manager open the diagnosis, handing the incident-command role to a senior engineer so that they can step back.

The review takes place two days later. The product lead, who handled the customer complaints, opens by asking who approved the deployment. The manager redirects plainly.

The review is about how the system let a change reach production without the failing path being tested, why the rollback took twelve minutes when the tooling should allow four, and why the first update took eight minutes. Those questions produce three actions with owners: a contract test for the failing path, a rollback shortcut, and a runbook change that names the commander before the manager has to. The deploying engineer's name appears in the timeline, because it is a fact, and nowhere else.

The manager does not find the bug themselves in minute nine, though they might have managed it. Doing so would have taught the team that incidents are run by whoever knows most, when they should be run by a structure that still works while that person is asleep.

9. Related competencies

- CC-1 Psychological safety and productive conflict: the safety that makes a blameless review real, and the reaction to bad news that keeps incidents reported.
- CC-5 Values, integrity, and ethical judgment under pressure: obeying the error budget when it is inconvenient; the costly right call after a public failure.
- DE-4 Managing scope, risk, and dependencies across workstreams: ownership gaps and systemic risk that surface as incidents.
- TJ-6 Security, privacy, and compliance as engineering responsibilities: security incidents, which follow the same structure with different stakes.
- SV-6 Organisational design and leading change: the structural causes of orphaned systems and chronic boundary failures.

10. Self-check

1. Why mitigation before root cause? *Answer:* Customers are down and the cause can wait. The commander takes a few seconds to capture whatever evidence the restoration will destroy, then restores. Diagnosis begins once service is back.
2. What has a commander who starts debugging done? *Answer:* Left the role empty. Coordination and communication stop, and the incident goes back to everyone debugging with nobody in charge.
3. A senior stakeholder demands that the postmortem name who is at fault. How do you hold the line, and how is blamelessness different from no accountability? [org] *Answer:* Run the review on how the system failed and what to change, and say plainly that naming people is not its purpose. Blamelessness makes honesty about failure safe, and that is what makes the system fixable. Patterns of recklessness are a performance matter, handled separately and outside the review.

4. After a reorganisation the same fragile area keeps causing incidents and no team owns it. What is the fix? [org] *Answer:* Assign explicit ownership and fund the stabilisation. Orphaned systems do not heal, and every incident in one is an ownership decision put off.
5. What does the DORA research say about the relationship between delivery speed and stability? *Answer:* At the level of the delivery system they improve together. The organisations with the highest throughput also had the lowest change failure rates and the fastest recovery, so at that level speed and reliability are not traded against each other.
6. Why does the executive's behaviour after a public failure matter more than the incident policy? [exec] *Answer:* Everyone watches what happens after the worst incident, and that moment sets the culture. An executive who models blamelessness under maximum pressure, including owning their own contributing decisions, writes the real policy. One who hunts for a culprit erases the written one.
7. What is an error budget only as good as? [exec] *Answer:* Leadership's willingness to obey it when it is inconvenient. Shipping through an exhausted budget in front of the organisation teaches everyone what the rule is in practice.

Sources

- Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy (editors), Site Reliability Engineering: How Google Runs Production Systems (2016): incident management, service-level objectives, error budgets, postmortem practice.
- John Allspaw, Blameless PostMortems and a Just Culture (2012): the case for blameless reviews and a just culture.
- Nicole Forsgren, Jez Humble and Gene Kim, Accelerate: The Science of Lean Software and DevOps (2018): the research on throughput and stability improving together.

DE-4 Managing scope, risk, and dependencies across workstreams

seeing the whole, naming the risk, negotiating the boundary

1. Definition and why it matters

Slow coding causes few delivery slips. Most come from unmanaged scope, from risks that arrive as surprises and dependencies nobody confirmed, and, as scope widens, from status that has stopped being true. The competency is seeing the whole of a piece of work, beyond the part in front of you. That means defending scope by naming each addition and trading it against the date, making risk explicit and attacking the most uncertain items first, treating a dependency the other party has not confirmed as something short of a plan, and matching the amount of deliberation to how reversible a decision is. At organisational scope it turns into managing an information system, because the facts arrive filtered. A team can be fast, well estimated and well prioritised and still miss, because of the addition nobody named, the risk that stayed in someone's head, or the other team that was assumed to deliver to a timeline it never agreed. The examinations test this because the failures stay quiet until suddenly they do not: the masked slip, the watermelon status, the programme with no owner.

2. Core principles

1. **Name scope additions and trade them.** Scope creep is small yeses piling up unnoticed. Each addition is named, its cost is stated, and it is traded against the date or against something else. It is never absorbed.
2. **Make risk explicit and attack the biggest unknown first.** A risk kept in someone's head turns into a surprise. Write it down, size it, and de-risk the item with the highest uncertainty before the ones that are merely hard.
3. **An unconfirmed dependency is not a plan.** Until the other team has agreed the timeline, the plan contains a hope. Confirm early, track it where people can see it, and prepare a fallback.
4. **Match deliberation to reversibility.** The typical misallocation is agonising over reversible calls while rushing irreversible ones. Spend the scrutiny where the door is one-way.
5. **Fix boundaries with ownership and contracts, not with meetings.** Clearer ownership, explicit interfaces and fewer shared surfaces reduce coordination cost. Adding coordination meetings mostly turns engineering time into calendar time.

6. **Go to ground truth when status smells wrong.** Above team scope, managing risk is really managing the reporting system. Green keeps meaning green when status is defined by objective criteria, reviews examine evidence instead of colour, and people can see that bad news is safe to bring.

3. Models and evidence

The models in this unit are practitioner tools with clear provenance. One is an old observation with an empirical literature behind it, which the unit grades as practice in its management form.

The pre-mortem [practice]

Gary Klein, Performing a Project Premortem (2007) describes the technique. Before committing to a plan, the team assumes it has failed and writes down why. This makes dissent legitimate, brings out the risks that optimism was holding down, and costs far less than the postmortem it prevents. It produces candour only where safety already exists. In a low-trust room it produces the same silence as every other meeting.

One-way and two-way doors [practice]

This is Jeff Bezos's classification in Jeff Bezos, 2015 Letter to Shareholders (Amazon.com, Inc.) (2016), used here to decide where scrutiny goes. Irreversible decisions, such as data models, public contracts and vendor lock-in, deserve deliberation. Reversible ones should move fast. People often misjudge reversibility, and "temporary" choices harden into permanent ones, so the rule of thumb is to treat a door as one-way until it has been shown to be otherwise.

Conway’s law [practice]

Melvin E. Conway, How Do Committees Invent? (1968) observed that a system's structure mirrors the communication structure of the organisation that built it. This unit uses it as a diagnostic. When delivery friction keeps recurring at the same boundary and survives repeated process fixes, it is often the organisation's structure presenting a bill, and the lasting fix may be organisational design rather than another working group. TJ-1 Evaluating architectural decisions and their organisational consequences and SV-6 Organisational design and leading change cover the empirical support and the consequences for design.

Contract at the boundary [practice]

This practice, given its team-level form in Matthew Skelton and Manuel Pais, Team Topologies: Organizing Business and Technology Teams for Fast Flow (2019), makes the agreement between a service and its consumers explicit. It consists of an interface, a versioning and deprecation policy, and automated contract tests that catch a breaking change before integration does. It is the technical half of "ownership and contract at the boundary". It verifies only the contract that was written down, and it cannot create interface discipline where there is none.

Ground truth and status criteria [practice]

This organisational practice has no formal name, and the unit treats it as central above team scope. Status is defined by objective criteria in place of sentiment: what green means, what triggers red, and what escalation each state requires. Reviews examine evidence instead of colour, and leaders keep more than one channel to reality. When it is missing, every team reports green while the programme is visibly late. In that picture the teams are not what is broken. The reporting system is, because it has learned to optimise for reassurance.

---

4. Practice

The scope ledger

Every addition to a committed piece of work is written down with its cost and the trade that paid for it, such as a moved date or a deferred feature. With a ledger, "the requirements kept changing" becomes a list of decisions instead of a complaint, and the ledger is the artifact that stops additions piling up unnoticed.

The risk list, ordered by uncertainty

This is a short, living list of what could go wrong. Each item carries its likelihood, its impact, and the action that would reduce the uncertainty. The list is ordered by uncertainty, and severity does not set the order, because the most uncertain items are the ones that surprise. It is reviewed weekly, because a list nobody reads reduces no risk at all.

Dependency confirmation

For each dependency on another team or a vendor, record who agreed it, when, to what date, and what the fallback is if it slips. An entry with "assumed" in the first column is a risk and moves to the risk list. Vendors are treated the same way as internal teams, with less trust.

The pre-mortem at the start

At the start of any significant initiative, the team spends an hour assuming the initiative has failed and writing down why. What comes out seeds the risk list with the things optimism would have hidden.

The cross-team delivery review

This is a regular review across teams. Its purpose is to surface risks and dependencies while they are still cheap to deal with, and performing status is no part of it. It examines evidence: what was meant to be true by now, and whether it is. The test of its health is whether it ever produces bad news. A review that is always green is a ceremony.

---

5. Scaling note

At team scope the object is the scope, risks and dependencies of one workstream, and the manager keeps the ledger, the list and the confirmations personally. At organisational scope the object becomes the dependencies between teams and the risks that fall between them, since each team's view stops at its own boundary. The first job is to get to ground truth when status smells wrong. The second is to fix boundaries with ownership and contracts in place of meetings, and a maintained dependency map takes the place of hope. At executive scope the object is systemic risk: the dependencies that no single team's view contains, the programme that spans the organisation and has no owner, the vendor or person the whole business quietly depends on. True information is the executive's scarcest commodity. They secure it by engineering the reporting system, and by checking critical programmes against evidence and not against reassurance. How Judgment Scales sets out the pattern, and SV-6 Organisational design and leading change treats chronic boundary friction as a structural signal.

---

6. Judgment

- Names scope additions and trades them against the date instead of absorbing them. [team]
- Keeps an explicit risk list and attacks the biggest unknown first. [team]
- Confirms cross-team dependencies with the other team, early, and tracks them. [team]
- Matches deliberation to reversibility.
- Failure mode: silently absorbing additions. [team]
- Failure mode: keeping risks in your head until they materialise. [team]

- Failure mode: assuming another team will deliver on your timeline. [team]
- Failure mode: agonising over reversible calls while rushing irreversible ones. [team]
- Goes to ground truth directly when status smells wrong, stabilises the delivery, then addresses why the truth felt unsafe to report. [org]
- Fixes two teams that repeatedly break each other with ownership and contract at the boundary (interfaces, contract tests, clear responsibility) and not with mandated goodwill. [org]
- Maintains a shared dependency map across teams, reviewed on a cadence. [org]
- Treats concentration of critical context in one person as a risk to engineer away through pairing, documentation and rotation, before it hurts. [org]
- Handles a vendor slip by establishing impact and mitigation options first, then escalating with the vendor and communicating the revised picture. [org]
- Reads duplicated work discovered late as a missing shared view of what is being built, and not as a communication failure. [org]
- Failure mode: managing by status report. [org]
- Failure mode: punishing the masked slip before stabilising the delivery. [org]
- Failure mode: prescribing "more communication" for a boundary failure. [org]
- Failure mode: celebrating the hero who is also the single point of failure. [org]
- Engineers the reporting system so that green means green: objective status criteria, evidence-based reviews, visible safety for bad news. [exec]
- Verifies a critical programme against evidence when a confident leader keeps reassuring them, because reassurance is not data. [exec]
- Forces a single accountable owner with authority across the boundary onto cross-organisation work. [exec]
- Fixes an incentive conflict between two directors' organisations at the level that owns the incentives, instead of urging them to cooperate against their own goals. [exec]
- Reads chronic friction at the same boundary as a structural signal and considers organisational design as the fix. [exec]
- Failure mode: running the organisation on watermelon status, green outside and red inside. [exec]
- Failure mode: accepting a confident leader's word for a programme you have independent reason to doubt. [exec]
- Failure mode: letting cross-organisation initiatives drift ownerless. [exec]
- Failure mode: process-patching a boundary the organisation chart itself created. [exec]

7. Tensions

Scope discipline versus responsiveness. If every addition is refused, the work is irrelevant by the time it ships. If every addition is accepted, it never ships. The ledger resolves this, because additions are accepted as trades with a visible cost, neither refused nor absorbed.

Risk visibility versus alarm. A long, honest risk list can make a team look as if it is in trouble, while a short, reassuring one hides the surprise. The judgment lies in the ordering and the framing. Risks that have actions attached and are reviewed and retired make visibility read as control, and stop it reading as panic.

Trust versus verification. Going to ground truth on a team's status can feel like distrust of its manager, and accepting a confident report can be how a two-month slip stays hidden. The way through is to make verification a property of the system, applied to every programme, so that it is no judgment on a person. And the person who masked a slip is asked why the truth felt unsafe before they are asked anything else.

Contracts versus goodwill. Explicit interfaces and contract tests feel bureaucratic between teams that get along, and they are what is left once the people who got along have moved on. The judgment is in knowing where a contract earns its cost (at boundaries with many consumers or a history of breakage) and where goodwill is enough.

Ownership versus autonomy. Forcing a single owner onto cross-organisation work takes authority away from the organisations it spans. Leaving it without an owner preserves their autonomy and guarantees drift. The executive decides, and the failure lies in not deciding.

8. Worked scenario

A senior manager runs a programme that spans four teams and is three weeks from a committed launch. In a corridor conversation an engineer mentions in passing that their team's piece is "probably two months out". For six weeks that team's status in the weekly programme review has been green. Its manager is competent and well liked, and has said nothing.

There are two reflexes. One is to confront the manager about the masking, and the other is to escalate the slip immediately on the information available. The first deals with the wrong thing first, and the second escalates a rumour. Neither brings the launch under control.

The senior manager goes to ground truth. Within a day they have looked at the team's real state (the work remaining, the integration tests not yet passing, the dependency on another team that was never confirmed) and have a real estimate. It is six weeks, short of the rumoured two months, and only if the dependency lands. The status was wrong and so was the corridor, and finding the difference was worth a day. They then re-baseline the programme. The committed launch cannot hold at full scope, and the stakeholders hear this the same week with three options: move the date by six weeks, launch on the date with

the fourth team's piece descoped, or launch on the date with a reduced rollout that does not depend on it. The stakeholders choose. The cross-team dependency review exists to catch exactly this, so the senior manager examines why it did not. It had been reporting colour instead of evidence, and they change it so that each team states what was meant to be true by now and whether it is.

Only once the delivery is stable do they turn to the manager. The question they ask is "what made the truth feel unsafe to report", and they do not ask "why did you hide this". The answer is that the last team to report a slip in the programme review had been publicly pressed for a recovery plan on the spot. That is a fact about the senior manager's own review, and they change how it runs. The masking still earns the manager feedback, given as situation, behaviour and impact, and it comes second.

The senior manager does not manage by the status report in either direction: they neither trust the green nor act on the corridor. Both would have been a substitute for looking.

9. Related competencies

- DE-2 Estimation, planning, and honest commitments: the range that widens when a risk materialises, and the re-forecast that follows.
- DE-1 Prioritisation under constraint: the shared bottleneck as a dependency, and the single ranked view that manages it.
- SV-6 Organisational design and leading change: organisational design as the durable fix for chronic boundary friction.
- PL-2 Feedback and difficult conversations: the feedback owed to the manager who masked a slip, and the reaction to bad news that decides whether it is reported next time.
- TJ-1 Evaluating architectural decisions and their organisational consequences-
  : architecture and the coordination cost it creates; Conway's law in its design form.

10. Self-check

1. Why is an unconfirmed dependency not a plan? *Answer:* Until the other team has agreed the timeline, the plan contains a hope where a commitment should be. Confirm early, track it visibly, and prepare a fallback. An entry marked "assumed" belongs on the risk list.
2. What should a risk list be ordered by, and why? *Answer:* Uncertainty, not severity. The most uncertain items are the ones that surprise, and de-risking them first narrows the plan fastest.

3. A team masked a two-month slip in a four-team programme three weeks from launch. What do you do first, second, and why in that order? [org] *Answer:* First, get ground truth directly and re-baseline the plan with options for stakeholders, which stabilises the delivery. Second, address the masking, first as a question about why the truth felt unsafe to report and then as feedback. Delivery comes first because the choices are disappearing. The masking comes second because it says as much about the reporting system as about the person.
4. Two teams repeatedly break each other's integrations and blame flows both ways. What is the root-level fix, and what is the fix that does not work? [org] *Answer:* Ownership and contract at the boundary: explicit interfaces, contract tests, clear responsibility. Mandating more communication or more coordination meetings turns engineering time into calendar time and leaves the boundary as it was.
5. Every team reports green and the flagship programme is visibly late. What exactly is broken, and why will better dashboards alone not fix it? [exec] *Answer:* The reporting system, which has learned to optimise for reassurance. Dashboards display what the system reports. The fix is objective status criteria, reviews that examine evidence instead of colour, and visible safety for bad news, so that what goes in becomes true.
6. A critical programme is in trouble by every indirect signal and the responsible vice-president keeps reassuring you. What do you do? [exec] *Answer:* Respect the vice-president and verify anyway by going to look at the evidence. Reassurance is not data, and an executive keeps more than one channel to reality.
7. Delivery friction keeps recurring at the same boundary despite three process fixes. What should you consider? [exec] *Answer:* That the structure is the problem. Chronic friction at a boundary is often the organisation's design presenting a bill, and the lasting fix may be organisational design rather than another working group.

Sources

- Gary Klein, Performing a Project Premortem (2007): the pre-mortem technique.
- Jeff Bezos, 2015 Letter to Shareholders (Amazon.com, Inc.) (2016): one-way and two-way doors.
- Melvin E. Conway, How Do Committees Invent? (1968): systems mirror the organisations that build them.
- Matthew Skelton and Manuel Pais, Team Topologies: Organizing Business and Technology Teams for Fast Flow (2019): team boundaries, interaction modes and explicit contracts between teams.

DE-5 Designing how work flows

process selection, developer experience, and AI-assisted workflows

1. Definition and why it matters

Process exists to serve delivery, and never the reverse. Version 1.0 framed this competency as choosing the lightest process that works. Version 3.0 widens it to designing the flow of work as a whole: the ceremonies and cadences a team runs, the developer platform and tooling it works through, and the AI assistance that is now part of how code is written, reviewed and shipped. These belong together as one object for two reasons. They fail in the same way, through structure adopted because it is standard when it solves no named problem, and through tooling that is imposed on a team without being designed for it. And the diagnostic question is the same for all of them: what problem is this solving, and is it still solving it? The competency matters because process is easy to add and hard to remove, because developer experience is where a large share of an organisation's engineering capacity is quietly spent or saved, and because AI assistance changes review, testing and ownership whether or not anyone has decided how. The examinations test it through the zombie ceremony, the velocity that falls while everyone is busy, the standard applied to the inside of teams while the boundaries between them stay chaotic, and the planning cycle whose output is already stale.

2. Core principles

1. **Choose the minimum structure that solves the actual problem.** Use enough structure to coordinate and to create predictability, and stop before it becomes ceremony. The bias is towards less, and structure grows with the situation, not with fashion.
2. **Every ritual answers the question: what problem is this solving now?** A practice that no longer earns its cost gets pruned. Adopting, adapting and dropping practices are all routine.
3. **Standardise where teams meet; leave the inside to the team.** Interfaces, a consistent floor for what "done" means where work crosses a boundary, and a shared language for reporting are the proper subjects of standards. Tools, ceremonies and style inside a team belong to the team.
4. **Throughput problems are usually flow problems.** A team that is busy and ships nothing is suffering from fragmentation, interruption and too much in flight. The fix lies in the flow of work, and asking the people for more effort will not supply it.
5. **Developer experience is capacity.** The friction between an engineer and a shipped change (build times, environments, review latency, tooling) is a cost paid on every

change. Investment in it is investment in throughput, measured by what teams can now do.

6. **AI assistance is designed into the workflow, or it happens to it.** What it changes about review, testing and ownership is decided explicitly. Its effect is measured honestly, including when the honest answer is that it is not helping.

---

3. Models and evidence

This unit covers more contested ground than most, because the evidence on AI assistance is recent and mixed. The unit says so and does not pick the convenient studies.

Team topologies and cognitive load [practice]

The framework in Matthew Skelton and Manuel Pais, Team Topologies: Organizing Business and Technology Teams for Fast Flow (2019) holds that teams have limited cognitive load, that the shape of a team and its modes of interaction with other teams should be designed around that limit, and that a platform team exists to reduce the load on the teams that ship product. What this unit takes from it is the design stance. How work flows is something to be designed, and the platform is part of that design, not infrastructure that happens to exist.

Aligned autonomy [practice]

This is the organising principle for deciding what to standardise across teams. Direction, priorities and boundary standards are set centrally, and methods are left to the teams. Standardise where teams meet (interfaces, the floor for "done" where work crosses a boundary, the vocabulary of status) and leave tools, ceremonies and style inside a team to that team. It has no single author. Organisations converge on this position after trying both extremes. Its limit is that the line between "where teams meet" and "team internals" is itself a judgment. That line is redrawn when a boundary fails, and a difference between two teams' internal practices is no reason to move it.

Developer experience [practice]

The framework in Abi Noda, Margaret-Anne Storey, Nicole Forsgren and Michaela Greiler, DevEx: What Actually Drives Productivity (2023) distils its authors' earlier research, one of whom, Nicole Forsgren, had led the DORA programme. It identifies three dimensions that drive productivity as engineers experience it (feedback loops, cognitive load and flow state) and argues that all three can be measured and improved. The paper proposes the framework and reports no new empirical study of its own. In practice this means investment in developer experience can be measured by the friction it removes, and the amount of tooling installed is the wrong measure.

Goodhart's law [practice]

In the form given by Marilyn Strathern, 'Improving Ratings': Audit in the British University System (1997), the law says that when a measure becomes a target it ceases to be a good measure. For this unit it explains how healthy velocity numbers can sit alongside flat outcomes. The metric is measuring activity, and the cadence built on it is optimising for the wrong thing. SV-5 Engineering metrics and measurement gives metrics their general treatment.

AI-assisted developer productivity [contested]

There are two kinds of evidence, and so far they do not agree. In the controlled experiment reported in Sida Peng, Eirini Kalliamvakou, Peter Cihon and Mert Demirer, The Impact of AI on Developer Productivity: Evidence from GitHub Copilot (2023), developers completed a defined task substantially faster with an AI coding assistant. In the randomised trial reported in Joel Becker, Nate Rush, Elizabeth Barnes and David Rein (METR), Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity (2025), experienced developers working in their own mature repositories were slower with AI tools than without them, while believing they were faster. The studies measured different things, one a well-specified task and the other real work in a complex codebase. The honest position for a manager has three parts: the effect depends heavily on the work, the codebase and the engineer; self-reports of speed-up are unreliable; and the effect has to be measured under the team's own conditions. Two things are not contested. The tools change what review has to establish, which is the subject of TJ-3 Code review and quality standards, and they bring risks that need governance, covered in TJ-5 AI-assisted engineering.

The operating cadence [practice]

This is a practice with no formal name, and at executive scope the unit treats it as the organisation's process: the rhythm of planning, review and escalation through which the organisation senses what is happening and responds. A sound cadence is light and rolling against stable priorities. It includes a review of the portfolio based on evidence, has explicit

escalation paths with known triggers, and reserves time for correction. Its failure is planning theatre, where weeks of ceremony produce a plan that is stale within a month. The principle behind it is that at scale, responsiveness improves execution and more detailed central planning does not.

4. Practice

The ritual audit

Twice a year the team lists every recurring meeting and practice it runs, with the problem each one solves. Anything that has no current problem to solve is dropped for a month, and if nobody notices, it stays dropped. The exercise is cheap, and it usually shows that a third of the calendar was ceremony.

The flow review

When throughput drops, the manager looks at the flow before looking at the people. How many things are in progress? How many are blocked, and on what? How much of each engineer's week goes on interruptions? The result is a change to the flow, such as a work-in-progress limit, a fixed interruption rota, or a batch of blocked work cleared in one go. It is not a request for more effort.

Standards at the seams

A group of teams keeps a short written list of what is standard because work crosses boundaries (interface conventions, the floor for "done", the reporting vocabulary), together with an explicit statement that everything else belongs to each team. The list is reviewed when a boundary fails. Differences between one team's internals and another's do not trigger a review.

Measuring developer experience

At a fixed cadence, the friction between an engineer and a shipped change is measured in the engineers' own terms: build and test times, environment setup, review latency, and a short survey on feedback loops, cognitive load and flow. Investment is prioritised by the friction it removes and reported the same way.

Introducing AI assistance deliberately

When AI tooling arrives, the team decides three things in writing: what review must now establish for generated code, who owns a change whatever produced it, and how the effect

will be measured in the team's own conditions. Adoption is not the measure, and neither is the number of licences in use. The measure is whether the team can now do something it could not do before, at a quality it can defend.

5. Scaling note

At team scope the object is the team's own process and tools, which the manager adopts, adapts and prunes directly. At organisational scope the object becomes the balance of consistency and autonomy across teams (what is standard because work crosses boundaries, and what is left inside them), along with the developer platform as a department-scale investment measured by what teams can now do. At executive scope the object is the organisation's operating system: the cadence of planning, review and escalation, the standards at organisational boundaries, and the investment in developer experience and AI-assisted work as strategic bets. The executive designs the system that does the executing and does not execute personally. How Judgment Scales sets out the pattern, and SV-6 Organisational design and leading change covers the structure the flow runs through.

6. Judgment

- Adopts a practice to solve a named problem and prunes it when it stops earning its cost. [team]
- Adapts frameworks to the team instead of following them dogmatically. [team]
- Reads a throughput drop as a flow problem (interruptions, fragmentation, too much in flight) and changes the flow instead of pushing the people. [team]
- Introduces AI tooling with explicit decisions about review, ownership and measurement. [team]
- Failure mode: heavyweight frameworks adopted because they are standard. [team]
- Failure mode: zombie ceremonies. [team]
- Failure mode: equating more process with more rigour. [team]
- Failure mode: big-company process on a tiny team. [team]
- Failure mode: an AI rollout measured by adoption, or by self-reported speed.
- Standardises where teams meet (interfaces, a floor for "done" at the seams, shared reporting language) and leaves team internals alone. [org]
- Judges cross-team rituals by the risks they surface, and not by the attendance they command. [org]
-

Invests in developer experience at department scale, measured by friction removed. [org]

- Governs AI-assisted work across teams proportionately, and reads where it is eroding understanding instead of adding leverage. [org]
- Failure mode: standardising team internals while boundaries stay chaotic. [org]
- Failure mode: meeting inflation as the answer to every coordination failure. [org]
- Failure mode: a definition of done nobody enforces at the seams where it matters. [org]
- Runs a light, rolling planning cadence against stable priorities, with evidence-based portfolio review and explicit escalation triggers. [exec]
- Sets direction, priorities and boundary standards centrally and leaves methods to teams. [exec]
- Measures delivered outcomes, and recognises a dashboard of healthy activity over flat outcomes as measuring motion. [exec]
- Decides the organisation's investment in developer experience and AI-assisted work as a strategic bet and reports its effect honestly. [exec]
- Builds the system that executes instead of executing personally, and holds directors accountable for execution. [exec]
- Failure mode: planning theatre, weeks of ceremony for a stale artifact. [exec]
- Failure mode: centralising methods instead of direction. [exec]
- Failure mode: the executive as the organisation's most senior programme manager. [exec]

7. Tensions

Structure versus lightness. With too little structure a growing team loses predictability, and with too much it loses the time the structure was supposed to protect. The judgment lies in scaling structure to the situation, and in being willing to take out what was added once the situation changes.

Consistency versus autonomy. Every standard imposed on a team's internals costs the team something and buys the organisation something, and most such standards buy less than they cost. Drawing the line at the boundary resolves this, and the line is redrawn only when a boundary fails.

Investment versus delivery. Developer-experience work competes with features for the same people, and its return comes later and is harder to see. The judgment is in measuring the friction so that the return becomes visible, and in protecting the investment against the urgency of any one quarter.

Adoption versus understanding. AI assistance can raise output while lowering the team's understanding of its own systems, and the second effect takes longer to show than the first. The answer is to design the review and ownership rules before adoption, and to measure understanding as well as throughput.

Planning versus responsiveness. A detailed plan is reassuring and goes stale within a month. A rolling cadence is harder to communicate and stays true. At scale the evidence favours responsiveness, and the friction comes mostly from stakeholders who want the comfort of the detailed plan.

8. Worked scenario

A head of engineering inherits an organisation where quarterly planning takes up three weeks of every quarter. Each team writes a detailed plan, the plans are reconciled over a series of long meetings, and the result is a document that leadership approves with some ceremony. Within a month it is stale, because priorities shift, dependencies slip and nobody has time to re-plan. Teams have learned to plan defensively. Stakeholders treat the document as a contract and engineers treat it as fiction. The chief executive likes the document and has asked for the next cycle to be more thorough.

The temptation is to comply and polish the ceremony with a better template, a planning tool or a longer horizon. That only makes the theatre more expensive. The diagnosis is that the cadence is upside down: a large central planning effort set against unstable priorities, with no time reserved for correction. At this scale, more detailed central planning does not improve execution, and responsiveness does.

The head of engineering replaces the cycle instead of refining it. At portfolio level, priorities are set in a short review with the executive team, based on evidence, and held stable for the quarter. Teams plan against those priorities in rolling six-week horizons, which takes a day where it used to take three weeks, and they re-plan when something material changes. A monthly review examines evidence (what was meant to be true by now, and whether it is) and has explicit triggers for escalation. Standards are set at the seams: how teams state dependencies, what "committed" means, and the vocabulary of status. Inside those limits each team runs whatever process it finds works.

The head of engineering meets the chief executive's concern head on and in the chief executive's own terms. What the chief executive wants is confidence that the organisation knows what it is doing. The head of engineering shows that the old document delivered the feeling of that confidence without the substance, and that the new cadence will give a true picture every month in place of a false one every quarter. They commit to reporting delivered outcomes against the stable priorities, and say plainly that the first quarter will look less tidy.

The head of engineering does not take on the coordination personally. With the old cycle gone, the temptation is to become the person who knows everything. The design has directors owning execution within the cadence, while the head of engineering runs the system and leaves the programmes to them.

9. Related competencies

- DE-1 Prioritisation under constraint: the stable priorities the cadence plans against, and work-in-progress limits as flow.
- DE-2 Estimation, planning, and honest commitments: batch size and predictability; what "committed" means at the seams.
- TJ-3 Code review and quality standards: what review must establish for AI-generated code, and the quality floor at boundaries.
- TJ-5 AI-assisted engineering: the governance and risk of AI-assisted engineering.
- SV-5 Engineering metrics and measurement: measuring outcomes over motion; Goodhart's law in full.
- SV-6 Organisational design and leading change: the team boundaries and platform structure the flow runs through.

10. Self-check

1. What question should be asked of any ritual? *Answer:* What problem is this solving, and is it still solving it? A practice with no current answer is pruned. Process is easy to add and hard to remove, so the bias is towards less.
2. Velocity has dropped for three sprints; engineers say they are busy but nothing ships. What do you look at first? *Answer:* The flow, not the people: how much is in progress, what is blocked, and how much of each week goes on interruptions. The fix is a change to the flow, such as a work-in-progress limit, an interruption rota, or blocked work cleared in a batch. More effort is not the fix.
3. What belongs in an organisation-wide standard and what should stay team-local, and why is the boundary the answer? [org] *Answer:* Standards belong where teams meet: interfaces, a consistent floor for "done" where work crosses a boundary, and a shared reporting language. That is where one team's variation becomes another team's incident. Inside a team, tools, ceremonies and style belong to the team, because standardising them costs more than it buys.
4. How should a department-wide rollout of AI tooling be measured? [org] *Answer:* By what teams can now do, at a quality they can defend, in their own conditions.

Licences used, adoption rates and self-reported speed are the wrong measures, and the evidence shows self-reported speed is unreliable.

5. What do the two kinds of evidence on AI-assisted productivity say, and what is the honest management position? *Answer:* A controlled experiment on a defined task found a large speed-up. A randomised trial of experienced developers in their own repositories found a slowdown the developers did not perceive. The effect depends on the work, the codebase and the engineer, self-reports are unreliable, and the effect has to be measured in the team's own conditions.
6. Why does more detailed central planning fail at scale, and what should an operating cadence optimise for instead? [exec] *Answer:* The plan is stale before it is finished, and the ceremony uses up the capacity it was meant to direct. The cadence should optimise for responsiveness: stable priorities, rolling short-horizon plans, evidence-based review, explicit escalation triggers, and time reserved for correction.
7. What has an executive who personally drives the most critical programme stopped doing? [exec] *Answer:* Running the system that runs everything else. Their role is to set direction, keep the environment clear of blockers, and hold directors accountable for execution.

Sources

- Matthew Skelton and Manuel Pais, Team Topologies: Organizing Business and Technology Teams for Fast Flow (2019): team shapes, interaction modes, cognitive load, the platform team.
- Abi Noda, Margaret-Anne Storey, Nicole Forsgren and Michaela Greiler, DevEx: What Actually Drives Productivity (2023): the developer-experience framework of feedback loops, cognitive load and flow state.
- Marilyn Strathern, 'Improving Ratings': Audit in the British University System (1997): the standard phrasing of Goodhart's law.
- Sida Peng, Eirini Kalliamvakou, Peter Cihon and Mert Demirer, The Impact of AI on Developer Productivity: Evidence from GitHub Copilot (2023): the controlled experiment on AI-assisted task completion.
- Joel Becker, Nate Rush, Elizabeth Barnes and David Rein (METR), Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity (2025): the randomised trial on experienced developers' productivity with AI tools.

Strategy & Vision

What this domain is

Strategy & Vision is the work of connecting engineering effort to business outcomes and direction and, from organisational scope, of shaping the organisation that does the work. Here engineering management turns outward. The team and its work give way to the business and its goals. The domain carries 20% of every examination: 12 questions on EMA-I, 15 on EMP-II and 18 on EME-III. Its six competencies are listed below with the questions each carries at each level, and the Exam Content Outlines give the proficiency indicators.

The throughline

Each competency in this domain reasons about outcomes in preference to output. Strong managers tie every significant piece of work to why it matters to the business. They speak in the listener's language, make resource trade-offs explicit, measure whether the strategy is working, and treat the organisation's structure as the way the strategy is implemented. The examinations favour the option that links an engineering decision to an outcome, or to a choice a stakeholder really faces, over the option that follows engineering preference. The stronger answer states the trade-off where the weaker one absorbs the ask. It argues the platform in business terms, compares bets on forward-looking arithmetic, trends teams instead of ranking them, and changes the structure instead of blaming the people.

How the domain scales

At team scope you translate a strategy you were given into work one team can execute and believe in, and you are the team's interface to its partners. At organisational scope you own a department strategy with a connecting thesis. You negotiate every ask with its trade-off, curate a portfolio with admission criteria, allocate a budget, measure across teams without breaking them, and design team boundaries. At executive scope you author the engineering strategy and argue it to the board. You own the horizon mix and the economics of the whole function, declare how the strategy's success will be judged, and shape the organisation itself. The general pattern is in How Judgment Scales; each unit's scaling note places its own competency on it.

Reading order

Read SV-1 Engineering strategy first, because everything else serves the strategy. Take SV-3 Roadmapping and SV-4 Engineering economics together: the roadmap is the sequence of bets, and economics decides between them. SV-2 Communicating and partnering across functions is the communication that carries all of it outward, and it is worth reading alongside the trade-off transparency in DE-1 Prioritisation under constraint. SV-5 Engineering metrics and measurement closes the loop by asking whether the strategy is working. Leave SV-6 Organisational design and leading change for last. It is assessed from EMP-II and makes the domain's claim that the organisation is how the strategy gets implemented.

Across domains

Strategy depends on Delivery & Execution for the mechanics its choices turn into: DE-1 Prioritisation under constraint for the portfolio, DE-2 Estimation, planning, and honest commitments for the commitment and DE-5 Designing how work flows for the flow. It depends on Technical Judgment for the technical strategy, TJ-1 Evaluating architectural decisions and their organisational consequences, and for the AI decisions that now sit inside economics and metrics, TJ-5 AI-assisted engineering. The people the structure is designed around are the subject of PL-6 Leading through leaders. When someone asks for a forecast or a message to be shaded upward, the costly right call is CC-5 Values, integrity, and ethical judgment under pressure.

SV-1 Engineering strategy

translating business goals into technical direction, and writing it down

1. Definition and why it matters

Engineering strategy is the translation of business goals into technical direction that people can execute and reason about. It also covers the part v3.0 adds to the title: writing it down, so that it can be argued with, delegated against and checked. Strategy arrives in business language (grow revenue, reduce churn, enter a segment). A bad translation produces either a literal feature checklist with no theory of impact or a set of themes too abstract to act on. A good one produces outcomes that tell people what success looks like and leave them free to find a cheaper path than the one first imagined, joined by a thesis that explains why these things were chosen over others. An organisation without a written strategy is run by whoever spoke last. And a strategy is tested by whether an engineer three levels down can say why their work is on the plan, whatever was announced. The examinations test this through the roadmap that "doesn't add up to a strategy", the pivot that leaves dead work limping on, and the company plan that assumes capabilities engineering does not have.

2. Core principles

1. **Frame work as outcomes, not outputs.** An outcome is the change wanted in the world; an output is the thing to be built. Outcomes define success and leave the path open. Outputs do neither.
2. **Ask what would have to become true.** For each goal, the conditions for success organise the work. When reality shifts, derive the work again from those conditions instead of defending the list.
3. **A strategy is a diagnosis and a set of choices, including what you will not do.** A "strategy" that keeps every option open has decided nothing. So, usually, has one that everybody agrees with at once.
4. **Write it down.** Nobody can argue with, delegate against or check a strategy that exists only in conversation. The document is short. It states the diagnosis, the choices and the reasoning, and it is what survives the offsite.
5. **Communicate intent and boundaries, not solutions.** A strategy has landed when teams can make good local decisions without asking. Getting there takes a campaign: the strategy is repeated relentlessly and shows up in visible decisions about what gets funded and what gets stopped. One announcement does not do it.
6.

> **Translation runs both ways.** Engineering strategy derives from company strategy, and engineering reality informs company strategy. When the company's plan assumes a capability the organisation does not have, the manager's job is to say so early. Absorbing it in silence fails the company twice.

---

3. Models and evidence

Strategy has a large literature and little controlled evidence. The models below are practitioner frameworks with clear provenance, and are graded accordingly.

Outcome versus output [practice]

This is the habit of framing work as the change wanted in the world instead of the thing to be built. It is the central translation move of this unit and has no single author. A manager can test it by asking of any item on any plan: why is this here? If the answer ties to a business outcome, the translation worked. It has two limits. Some work is legitimately shaped as output, such as a contractual deliverable or a compliance requirement. And an outcome nobody can influence is a wish rather than a goal.

Strategy, vision, and roadmap [practice]

The distinction follows Richard P. Rumelt, Good Strategy Bad Strategy: The Difference and Why It Matters (2011). A vision names the destination and why it matters. A strategy is a diagnosis of the situation, a guiding policy that responds to it, and a coherent set of actions; in other words, a set of choices, including what will not be done. A roadmap is merely the current sequence of bets that follows. Rumelt's account of bad strategy (fluff, failure to face the problem, goals mistaken for strategy, and a wish list of objectives) is the most useful checklist a manager has for reviewing a strategy document. The choices matter more than the artifacts, and a brilliant strategy that nobody can recite governs nothing.

Writing the strategy down [practice]

Will Larson, The Engineering Executive's Primer: Impactful Technical Leadership (2024) sets out this practice for engineering leaders: a short written engineering strategy giving the diagnosis of the current situation, the policies that follow from it, and the concrete actions. It is reviewed on a cadence and revised when the signals call for it, whatever the calendar says. Writing it down makes the strategy resilient, because the outcomes and principles are stable enough to survive change while the means stay free to adapt. It also makes the strategy checkable, since a written choice can be compared with what was in fact funded.

The connecting thesis [practice]

This is an idea, not a named model, and this unit treats it as what separates a strategy from a wish list. The thesis states why these teams do these things, tying the work of several teams to a few department-level outcomes. Without it you get the bottom-up roadmap, which is team wish lists stapled together. Adding detail to the list never fixes that; the fix is to define the outcomes and derive the list again from them. The same idea sets the altitude of a manager's own goals. A manager of managers sets objectives at the level of department outcomes, and a roll-up of team task lists does not qualify.

---

4. Practice

The one-page strategy

A short document. It gives the diagnosis in a paragraph, the three to five choices the organisation is making with the reasoning for each, what it is explicitly not doing, and how it will know whether the strategy is working. It is reviewed against reality every quarter. If it will not fit on a page, the choices have not been made.

The why test

From time to time the manager picks an item from the current plan at random and asks the person doing it why it is there. An answer that connects to an outcome in the strategy means the translation is working. "Because it's on the roadmap" means the strategy has not landed, and the fix lies in communication rather than in the roadmap.

Re-deriving on a pivot

When the company changes direction mid-quarter, the strategy is derived again from the new outcomes within days. Work that is now irrelevant is killed explicitly instead of being left to limp on. Work that survives is justified afresh. The reasoning is over-communicated to the teams whose work has just evaporated. Dead roadmap items that go on consuming capacity are the signature of a badly handled pivot.

Strategy as campaign

The written strategy comes up in every planning conversation and is built into decisions about what to fund and what to stop. It is told as a memorable narrative, again and again, until it survives being retold three levels down. The check is whether a team lead can recite it. If they cannot, it has been announced without being communicated.

Surfacing the capability gap

When a company plan assumes something engineering cannot yet do (a scale, a security posture, a platform), the manager says so explicitly and early, and brings options: either the strategy adjusts or the capability is funded. Where the company has no clear product strategy, engineering brings options and evidence to force the conversation. It also makes plain that it is contributing to product direction and has no intention of replacing it.

5. Scaling note

At team scope the manager takes a given strategy and translates it into one team's work, framing outcomes, deriving features from a theory of impact, and making sure every engineer can say why their work matters. At organisational scope the object becomes a department strategy with a connecting thesis. It is communicated as intent and boundaries so that decisions can be made locally, and it is defended against churn and encroachment. The characteristic failure here is the stapled wish list. At executive scope the manager writes the engineering strategy itself, argues for it at the executive table and keeps it alive as the world changes. They also recognise that the organisation, meaning its structure, skills and architecture, is how the strategy gets implemented, which is the subject of SV-6 Organisational design and leading change. The general pattern is described in How Judgment Scales.

6. Judgment

- Frames the team's work as outcomes and derives features from a theory of impact. [team]
- Ensures every engineer can say why their current work matters to the business. [team]
- Works the plan out again when circumstances change instead of defending the original list. [team]
- Failure mode: strategy as a literal feature list. [team]
- Failure mode: themes too abstract to act on. [team]
- Failure mode: building things because they are on the roadmap. [team]
- Writes a department strategy with a connecting thesis that a team lead can recite. [org]
- Sets objectives at the altitude of department outcomes, not as a roll-up of team task lists. [org]
- Communicates intent, priorities and constraints so that teams decide locally without escalating. [org]
-

Re-derives fast on a pivot, kills newly irrelevant work explicitly, and over-communicates the reasoning. [org]

- Failure mode: stapling team wish lists together and calling it strategy. [org]
- Failure mode: broadcasting the strategy once and assuming it landed. [org]
- Failure mode: letting dead roadmap items keep consuming capacity after a pivot. [org]
- Writes an engineering strategy (diagnosis, choices, what it is not doing) and argues it at the executive table. [exec]
- Treats organisational design, skills and architecture as the strategy's implementation, and realigns them together. [exec]
- Tells the company what engineering cannot yet do before the plan depends on it, and forces the honest conversation. [exec]
- Fills a strategy vacuum with options and evidence developed jointly, avoiding both silence and a silent takeover. [exec]
- Communicates strategy through decisions about what is funded and what is stopped, and not only through decks. [exec]
- Keeps the strategy alive between planning cycles, revising it when signals warrant and not by the calendar. [exec]
- Failure mode: new strategy, old organisation chart. [exec]
- Failure mode: nodding along to a company plan the organisation cannot execute. [exec]
- Failure mode: treating the absence of product strategy as someone else's problem. [exec]
- Failure mode: a strategy that exists only in the offsite slides. [exec]
- Failure mode: a strategy that keeps every option open in the name of flexibility. [exec]

7. Tensions

Commitment versus flexibility. A strategy that commits will be wrong about some things, and one that hedges is no strategy at all. The answer lies in structure. Hold the outcomes and principles steady, leave the means free to adapt, and review on a cadence that changes the strategy when evidence demands it and not when nerves fail.

Direction versus autonomy. The more a strategy prescribes, the less teams can decide locally; the less it prescribes, the more they diverge. The line sits at intent and boundaries. The strategy gives the why, the priorities and what is not being done, and leaves the solutions to teams.

Honesty upward versus standing. Telling the executive team that the plan assumes a capability engineering lacks can sound like engineering saying no. Absorbing it in silence

sounds like agreement and fails later. What survives is a set of options with their costs, offered early.

Persistence versus responsiveness. A strategy lands through relentless repetition, and a pivot that contradicts last month's repetition costs credibility. The way through is to say plainly that the outcomes have changed and why, and to kill the old work visibly so the contradiction does not stand.

Engineering preference versus business outcome. The best technical direction is not always the one the business needs, and the examinations consistently favour the option that ties the decision to an outcome. The judgment lies in making engineering's case in the business's terms, and in accepting the answer when it goes the other way.

8. Worked scenario

A head of engineering reads the company's new annual plan a week before it goes to the board. It commits the company to a platform strategy, opening the product to third parties within a year, and it assumes an engineering organisation that can deliver this. The organisation cannot. It was built around a single product. The architecture has no stable external interfaces, the teams are organised by product feature, and nobody above the level of individual engineer has platform experience. The chief executive is enthusiastic, and the plan has been socialised with the board as settled.

The pull is to nod along, accept the plan as given, and try. That fails the company twice: the year will be lost, and the board will have been told something untrue. The opposite pull is to declare the plan impossible. That fails in a different way. It makes engineering the department of no, and it is also untrue, because the plan is possible if it is funded and sequenced.

The head of engineering translates in both directions. Downward, they write the engineering strategy that the company plan implies. The diagnosis is an organisation and an architecture built for one product. The choices are stable external interfaces first, a platform team, teams re-cut around capabilities instead of features, and two senior platform hires. The document also says what engineering will not do this year in order to do this. Upward, they take that document to the chief executive before the board meeting with three options. The first is the full platform in a year, which requires the re-cut, the hires, and stopping two committed product initiatives. The second is a staged version that opens one capability to a small set of partners in the first year and the rest in the second. The third is the original plan, with engineering's honest assessment that it will miss.

The chief executive chooses the staged version, and the board is told about a plan the organisation can execute, with checkpoints. The head of engineering then treats the organisation as the implementation. The team re-cut and the platform team are announced

together with the strategy, not afterwards, because a new strategy laid over the old structure lets the old structure become the real strategist.

They avoid both of the easy options: absorbing an impossible plan and hoping, or refusing it and being proved right in a year. The judgment lies in bringing the honest picture early, with options, in the company's terms.

9. Related competencies

- SV-2 Communicating and partnering across functions: arguing the strategy to executives and the board, and disagreeing on the merits.
- SV-3 Roadmapping: the roadmap as the current sequence of bets that follows from the strategy.
- SV-6 Organisational design and leading change: the organisation as the strategy's implementation, and realigning structure with direction.
- DE-1 Prioritisation under constraint: the portfolio the strategy governs, and stopping what no longer fits.
- TJ-1 Evaluating architectural decisions and their organisational consequences: the technical strategy as the architectural form of the same choices.

10. Self-check

1. What question tests whether a goal has been translated well? *Answer:* Pick any item on the plan and ask the person doing it why it is there. If the answer ties to a business outcome, the translation worked. If it is "because it's on the roadmap", the strategy has not landed.
2. What is the difference between an outcome and an output, and why does the difference free the path? *Answer:* An outcome is the change wanted in the world; an output is the thing to be built. Stating the outcome tells people what success looks like and lets them find a cheaper way than the one first imagined.
3. The executive team says your department roadmap "doesn't add up to a strategy". What is missing, and why does more detail not fix it? [org] *Answer:* A connecting thesis: a few department-level outcomes that explain why these teams do these things. Putting more detail into a stapled wish list only gives you a more detailed wish list. The fix is to define the outcomes and derive the roadmap again from them.
4. Half your roadmap became irrelevant with a mid-quarter pivot. What is the discipline? [org] *Answer:* Re-derive quickly from the new outcomes, kill the newly irrelevant work

explicitly instead of letting it limp on, and over-communicate the reasoning to the teams whose work evaporated.

5. Name Rumelt's three parts of a good strategy, and the sign that a "strategy" has decided nothing. *Answer:* A diagnosis, a guiding policy, and coherent action, which together form a set of choices that includes what will not be done. A strategy that keeps every option open, or that everyone agrees with at once, has decided nothing.
6. The company strategy assumes capabilities your organisation lacks. What is your responsibility, and what does silence cost? [exec] *Answer:* Raise the misalignment explicitly and early, with options: the strategy adjusts or the capability is funded. Silence means absorbing an impossible plan, which fails the company twice. The year is lost, and the board was told something untrue.
7. Why is announcing a new strategy over the old organisation chart a failure of strategy rather than of execution? [exec] *Answer:* Because the organisation (its structure, skills and architecture) is the strategy's implementation. If it is left unchanged, the old structure becomes the real strategist and the new direction stays on paper.

Sources

- Richard P. Rumelt, Good Strategy Bad Strategy: The Difference and Why It Matters (2011): diagnosis, guiding policy, coherent action; the anatomy of bad strategy.
- Will Larson, The Engineering Executive's Primer: Impactful Technical Leadership (2024): writing an engineering strategy and keeping it alive.

SV-2 Communicating and partnering across functions

product, design, executives, and non-engineering stakeholders

1. Definition and why it matters

A manager is the interface between engineering and the rest of the organisation: product, design, sales, finance, executives and customers. v1.0 framed the competency as communicating decisions and trade-offs outward. v3.0 widens it to partnering, because the relationship runs both ways. Product and design have constraints as real as engineering's. Executives decide better when engineering brings options instead of verdicts. And engineering's standing in a company depends on whether it shows up as the partner that prices choices or as the department that says no. The core skill is communicating in the listener's language and at their altitude, which means talking about outcomes, trade-offs and risks and leaving the architecture out. Its core move, at every scope, is to state the trade-off calmly when someone with power asks for something that breaks capacity or sequencing. Most of the situations that damage engineering's credibility are failures of communication: bad news surfaced late, an ask absorbed in silence until delivery fails, a flat refusal that comes across as obstruction. The examinations test this constantly, because the comfortable answer, yes or no, is nearly always the wrong one.

2. Core principles

1. **Speak the listener's language at the listener's altitude.** To a finance leader the same fact is cost and risk; to a product lead it is timeline and scope; to a customer it is impact and resolution. The job is translation, and a lecture means the translation failed.
2. **Bad news early, with options.** Raising a risk before it matures into a surprise does more to build stakeholders' trust than anything else a manager does. Set expectations you can beat, never the reverse.
3. **Every ask is answered with its trade-off.** Never with a bare yes or no. "We can do this, and here is what it displaces" lets the decision be made on the real price, and stops the manager absorbing demand until delivery fails.
4. **Partner sideways on shared outcomes.** Alignment with product and design comes from one plan, planned jointly, instead of two negotiated treaties. A conflict with a partner department is resolved at the level where priorities are owned, with the shared goal on the table.
5.

Bring options, not vetoes. Engineering stops being seen as a cost centre through how it behaves. Its default grammar becomes "yes, and here is the cost", and its work is visibly tied to revenue and risk.

6. **Disagree on the merits, then commit.** Argue directly against a strategy you believe is wrong: privately first, with evidence, framed on company outcomes. Escalate honestly if it matters enough. If the decision goes the other way, commit to it visibly instead of resisting quietly.

---

3. Models and evidence

Stakeholder communication belongs to practitioners, and the unit grades it that way. One of the models is a rule from a single company that has passed into general usage.

Communicating at the listener's altitude [practice]

This is a practice, not a named model, and it sits at the centre of the unit: the same update, re-cut for each audience's concerns and level of abstraction. The discipline is to start from what the listener has to decide and give them the outcome, the trade-off and the risk in those terms. Its characteristic failures are the one-size-fits-all update, the architecture lecture to a finance audience, and a reply to "move faster" that is either capitulation or a tutorial.

Trade-off transparency [practice]

DE-1 Prioritisation under constraint introduces this move, and here it becomes the core of negotiation. State what the request costs in terms of what else has to move, and let the person who wants it choose. At organisational scope a manager of managers uses it when asked to pull a strategic initiative forward, to absorb a new product line with no headcount, or to reprioritise every few weeks. The cost is named in numbers and the choice is made explicitly. The alternative is to absorb the request and then fail across the board.

Disagree and commit [practice]

The rule is stated in Jeff Bezos, 2016 Letter to Shareholders (Amazon.com, Inc.) (2017). When you disagree with a decision, say so directly and give your reasoning. If the decision goes the other way, commit to it fully instead of holding back effort while you wait to be proved right. In this unit it is how partnership works at the executive table. A peer's strategy you believe is wrong gets a direct, private disagreement based on evidence, then honest escalation if the stakes warrant it, then visible commitment. The rule assumes that a decision was in fact made and the disagreement was in fact heard. Used to silence dissent before the decision, it becomes a different rule, and a worse one.

Engineering as business partner [practice]

This is a stance, not a named model, and the unit treats it as the remedy for "the department that says no". Engineering brings options and trade-offs instead of vetoes, ties its work visibly to revenue and risk, and changes the system instead of fighting each instance. An example is an explicit gate for custom commitments, agreed with sales at executive level, in place of a fight over every deal. The evidence for it is experiential. Its failure is the executive who defends engineering with indignation where evidence was needed.

---

4. Practice

The audience-specific update

Before any update to a non-engineering audience, the manager writes one line: what does this person have to decide, and what do they need from me to decide it well? The update is then cut to fit (cost and risk, timeline and scope, or impact and resolution), and technical detail is available on request instead of delivered by default.

The early-warning habit

As soon as a risk to a commitment crosses the point at which the stakeholder would want to act, the stakeholder hears about it, with the probability, the cause and the options. They hear it before it is certain, and without excuses. This habit is what makes a later "we will beat this date" believable.

The cost-of-yes reply

Every request against a committed plan gets a reply of the same shape: yes, and this is what it displaces; or, this is what it would take. It is delivered calmly, in the requester's terms, and the choice stays with them. The reply is recorded where the next person to ask will see it.

Joint planning with product and design

There is one plan, produced together against shared outcomes, with dependencies made explicit. An engineering plan and a product plan reconciled by negotiation is the thing to avoid. When a partner department's roadmap conflicts with the department's own, the two sides look for a resolution together, at the level where both sets of priorities are owned and with the shared goal on the table, instead of letting the teams fight a proxy conflict.

Disagreeing at the executive table

STRATEGY & VISION

When a peer executive's strategy seems wrong for the company, the manager disagrees directly and privately first, with evidence, and frames the case on company outcomes instead of turf. If the stakes warrant it, they escalate honestly. Once the decision is made, they commit to it visibly. This practice exists to prevent the battle being fought through subordinates.

5. Scaling note

At team scope the manager deals with the team's product and design partners and with the stakeholder pressing for a date. The work is to translate, raise bad news early and set expectations that can be beaten. At organisational scope the counterparts become department-level partners and executives who need trade-offs explained. Communication turns into negotiation: the cost of every ask is stated in numbers, and sideways alignment rests on shared outcomes and joint planning. At executive scope the audiences are the board, the chief executive and peer executives, and engineering's standing in the company is at stake. Investment is argued as a staged business case, vague pressure is clarified before anyone answers it, and disagreement on the merits is followed by commitment. The pattern is in How Judgment Scales, and the economics behind the arguments are in SV-4 Engineering economics.

6. Judgment

- Pitches the same message to each audience's concerns: cost and risk, timeline and scope, impact and resolution. [team]
- Translates technical facts into business consequences. [team]
- Raises risks before they mature into surprises, and sets expectations to be beaten. [team]
- Leads with the trade-off and the reasoning when saying no or delivering disappointment.
- Failure mode: drowning non-engineers in detail. [team]
- Failure mode: going silent on problems. [team]
- Failure mode: one-size-fits-all updates. [team]
- Failure mode: over-promising to dodge a hard moment. [team]
- Answers every ask with its trade-off, calmly and in numbers, and gets the choice made explicitly. [org]
- Puts a price on priorities that change every few weeks, and negotiates a stable planning horizon with a fast lane for real emergencies. [org]

- Frames an investment plan to executives around business outcomes and the cost of not investing, instead of the feature list or the technology. [org]
- Builds alignment with product and design on shared outcomes and joint planning, producing one plan instead of two treaties. [org]
- Resolves a conflicting partner roadmap jointly, at the level where priorities are owned, instead of through a proxy war between teams. [org]
- Failure mode: heroically absorbing asks until delivery fails. [org]
- Failure mode: flat refusals heard as "engineering says no". [org]
- Failure mode: escalating partner conflicts as blame instead of resolving them as priority calls. [org]
- Failure mode: investment pitches framed in technology instead of outcomes. [org]
- Frames a multi-year investment to a sceptical board as business strategy: what it enables and protects, what not investing costs, and checkpoints for each stage. [exec]
- Answers "why did headcount grow without faster output" with what the scaling bought and with outcome measures, instead of a promise to make everyone busier. [exec]
- Clarifies and diagnoses "engineering needs to move faster" before defending or promising. [exec]
- Repositions engineering as the partner that prices options, with "yes, and here is the cost" as the default grammar. [exec]
- Changes the system with a peer executive, for example an explicit gate for custom commitments, instead of fighting sales deal by deal. [exec]
- Disagrees with a peer executive directly and privately, on evidence and company outcomes; escalates honestly; commits visibly once the call is made. [exec]
- Failure mode: defending engineering with indignation instead of evidence. [exec]
- Failure mode: answering "move faster" with capitulation or a lecture. [exec]
- Failure mode: letting "no" become engineering's public vocabulary. [exec]
- Failure mode: fighting peer-executive battles through subordinates. [exec]

7. Tensions

Candour versus standing. Raising a risk early can make engineering look like the bearer of bad news yet again. Hiding it protects standing until the moment it destroys it. Options resolve this, because bad news delivered with choices comes across as control instead of alarm.

Partnership versus capacity. Being the partner who says yes and being the manager who protects the team's capacity pull in opposite directions. The cost-of-yes reply honours both: the yes is real, and so is the price.

Simplicity versus accuracy. A message cut to the listener's altitude leaves out detail that matters to engineers, and a message carrying that detail loses the listener. The judgment lies in knowing which details would change the listener's decision, and keeping the rest available without delivering it.

Dissent versus commitment. Arguing hard against a peer's strategy and then executing it fully feels inconsistent. It is still the only stance that keeps both the argument and the organisation honest. Failure is possible on either side, as silence before the decision or as resistance after it.

System versus instance. Each custom commitment that sales makes is easier to fight than the system producing them, and the fights never end. The judgment is to spend political capital once, at executive level, on the gate.

---

8. Worked scenario

In front of two sales colleagues, a product lead asks a team's manager when the new reporting feature will ship. Last week the team estimated six to ten weeks. The width of the range comes from an unresolved dependency on the data platform and from a design question the product lead has not yet answered. The product lead wants a date, says a range is not something they can take to a customer, and suggests that "eight weeks" would be fine to say.

There are two easy answers. The manager could say eight weeks, which becomes a commitment as soon as it is spoken and will be defended against the evidence for two months. Or they could repeat that the team cannot give a date until the dependency is resolved, which is true and, in front of sales, makes engineering look difficult.

The manager translates instead. To the sales colleagues, in their terms, they say the team can commit today to delivering the first version, covering the two report types the customer asked for, within ten weeks, and will know within two weeks whether it can be sooner. Sales can take that sentence to the customer, and it is true. To the product lead, in their terms, they explain that the width of the range comes down to two decisions. One is the product lead's own design question. The other is a dependency the manager will confirm this week. An answer on the design question by Friday narrows the range by itself. The manager offers to take the product lead through the two drivers in ten minutes after the meeting, away from sales.

They also do the early-warning work as soon as they can. The data-platform dependency is confirmed that week: it will land in week five instead of week three, which moves the

range to seven to ten weeks. The product lead hears this on the day it becomes known, along with the option it creates, which is to hold the ten-week commitment or to descope one report type to protect eight. The product lead chooses to hold ten. The customer was never told eight.

The manager declined to give the number that would have made the meeting comfortable and turned the next two months into the defence of a date nobody believed. The honest answer cost a slightly harder conversation in front of sales. The other answer would have cost engineering its credibility with both.

9. Related competencies

- DE-1 Prioritisation under constraint: the queue the ask is placed in, and the cost of yes as prioritisation.
- DE-2 Estimation, planning, and honest commitments: the honest forecast a stakeholder is given instead of the date they asked for.
- SV-1 Engineering strategy: the strategy the department argues from, and the capability gap raised upward.
- SV-4 Engineering economics: the economics behind an investment argument, covering cost, headcount, and the price of not investing.
- CC-5 Values, integrity, and ethical judgment under pressure: declining to shade a message under executive pressure, and absorbing the cost.

10. Self-check

1. Why is surfacing bad news early a trust-building move rather than a damaging one? *Answer:* Because it arrives while the stakeholder still has options, and because a manager who reports risks early is believed when they say a date will be beaten. Late surprises turn delivery problems into trust problems.
2. The same delivery risk must be explained to a finance leader, a product lead and a customer. What changes? *Answer:* The altitude and the terms: cost and risk for finance, timeline and scope for product, impact and resolution for the customer. The fact stays the same, and the message is cut to what each of them has to decide.
3. You are told to absorb a new product line with no headcount. What does the strong response contain that a yes or a no does not? [org] *Answer:* The trade-off. It sets out what the department will stop or slow, in numbers, with a recommendation, and has the person who owns the priority make the choice explicitly. A yes absorbs the work and fails broadly; a no refuses the business.

4. Leadership changes priorities every few weeks and your teams are suffering whiplash. What do you negotiate? [org] *Answer:* Name the cost of the churn in lost throughput and eroding trust, and negotiate a stable planning horizon with an explicit fast lane for real emergencies. That keeps the organisation responsive without constant re-sequencing.
5. The chief executive says "engineering needs to move faster" with no specifics. What comes before defending or promising? [exec] *Answer:* Clarify and diagnose. Faster at what, by whose measure, compared with what? Then respond with a picture grounded in evidence and a concrete plan. Capitulation and a lecture are the two failures.
6. The board asks why headcount grew without obviously faster output. What makes a response credible? [exec] *Answer:* Showing what the scaling bought (reliability, security, the capacity to run more products, risk retired) with outcome measures, instead of promising to make everyone busier. Growth that is not measured cannot be told apart from bloat.
7. A peer executive is pushing a strategy you believe is wrong for the company. What is the sequence? [exec] *Answer:* Disagree directly and privately first, with evidence, framed on company outcomes instead of turf. Escalate honestly if it matters enough. If the decision goes against you, commit visibly; do not resist quietly or fight through subordinates.

Sources

- Jeff Bezos, 2016 Letter to Shareholders (Amazon.com, Inc.) (2017): disagree and commit.

SV-3 Roadmapping

balancing product delivery against platform and quality investment

1. Definition and why it matters

A roadmap is a sequence of bets about where to spend finite capacity. Its permanent tension is between shipping visible value now and investing in the platform and quality that make future value possible. Starve the platform and velocity decays under accumulating debt and toil. Over-invest and the business goes hungry for value it can see. The competency is holding the balance deliberately. That means allocating capacity explicitly so that foundational work is not deferred for ever, and treating the roadmap as a living instrument of trade-offs, framed around outcomes, instead of a list of dated promises. It also means separating near-term commitments from directional bets, and saying what is not being done. The platform never asks for capacity and the business always does, so the balance holds only when a manager holds it on purpose. The examinations test this through the product partner who wants no reduction in feature output, the roadmap that is the union of everyone's requests, the internal team funded by inertia, and the two-year bet started on borrowed patience.

2. Core principles

1. **Allocate capacity explicitly.** Set a standing share of the team's capacity for platform, reliability and debt, state it and defend it, so that foundational work is a decision and not whatever happens to be left over. Zero is a decision too, and it has a date on which it fails.
2. **The roadmap is bets, not promises.** Near-term commitments are firm. Directional bets carry confidence levels. Both are framed as outcomes. A roadmap presented as dated promises is a list of future disappointments.
3. **Argue foundations in business terms.** Platform investment is justified by what its absence costs in velocity, incidents and features that cannot be built. Engineering aesthetics are no justification.
4. **Admission is by outcome contribution.** An initiative gets onto the roadmap by its expected contribution to stated outcomes, weighed against cost and risk. Which team wants it, or which stakeholder asked last, does not count.
5. **Say what is not being done.** The non-goals tell a reader more than any other section of a roadmap, and they are the section most often missing.
6.

Long bets are staged, protected, and funded by subtraction. A multi-year platform investment needs intermediate value where possible, an honest estimate of the near-term sacrifice, secured commitment before it starts, and a defence against the quiet nibbling of "temporary" reallocations. An existential new priority is funded by explicitly clearing something out. Adding it to everyone's full plate does not fund it.

---

3. Models and evidence

Roadmapping is practitioner work. One of the biases underneath it has a research literature, and is graded accordingly.

Explicit capacity allocation [practice]

This is a discipline, not a named model, and the unit treats it as fundamental: a stated fraction of capacity for platform, reliability and debt, held under pressure. It has no author. Every organisation that keeps its platform alive has arrived at it. Without it the trajectory is predictable. Feature pressure consumes everything until velocity collapses, and then the platform gets a crisis budget that costs more than the standing share would have.

The three horizons [practice]

Mehrdad Baghai, Stephen Coley and David White, The Alchemy of Growth: Practical Insights for Building the Enduring Enterprise (1999) offers this framework for balancing a portfolio over time. It separates the current business, the emerging businesses that will matter in a few years, and the options that might disrupt everything. Applied to an engineering roadmap it becomes an executive question: how much capacity serves today's product, how much builds the platform the strategy needs in two years, and how much explores. The exact proportions depend on context. The value lies in making the mix of horizons an explicit decision, owned at executive level, so that it is not an accident of whoever shouted loudest this quarter.

Sunk-cost discipline [research]

Hal R. Arkes and Catherine Blumer, The Psychology of Sunk Cost (1985) found that people keep investing in a course of action because of what they have already spent, even when the forward-looking comparison says stop. In roadmapping this governs the half-done migration, the championed bet and the legacy team funded by inertia. Money already spent argues for nothing. The only thing that counts is the value remaining against the cost remaining, compared with the alternatives. The opposite error also happens: abandoning a sound long bet the moment a fashionable alternative appears. What separates the two

is that changed fundamentals justify switching and novelty does not. SV-4 Engineering economics covers the general case.

Staged bets and checkpoints [practice]

This is a practice, not a named model. A long investment is structured to deliver intermediate value where it can, to state its near-term cost honestly, and to include explicit checkpoints for doubling down or folding. That structure is how a two-year platform bet survives a quarter's urgency. It is also how a bet on a disruptive technology is sized to teach, since a bet too small to matter and one too big to survive being wrong are both errors.

---

4. Practice

The stated balance

The team's capacity split between features, platform and reliability, and debt is written down as percentages, agreed with the product partner and reviewed each quarter. When pressure arrives, the conversation is about changing the number, explicitly and for a stated period. Whether the platform work can wait just this once is not the question on the table.

The investment case in business terms

Any platform investment larger than a sprint is argued as a business case. It covers what the absence of the work costs now (incidents, cycle time, blocked features), what it enables, and what happens if the current trajectory continues. It uses numbers where they exist and names its assumptions where they do not. Engineering elegance has no place in it.

Admission criteria

A department roadmap comes with a written statement of how initiatives get onto it: expected contribution to stated outcomes, cost, risk and dependencies. A request that does not meet the criteria is answered by pointing to the criteria, which is different from a no.

Reading over-commitment from flow

The signs are mechanical and are checked monthly. Everything is in progress and nothing is finishing; there is no capacity for the unplanned; dates are missed in bulk; there is no slack for discovery. If any of them is present, the answer is to cut work. Exhortation will not help.

The mandate check

Every standing investment, such as an internal tools team or a long-running platform group, should be able to state its strategic value and how its success is measured. One that cannot is assessed against the strategy. It gets a clear mandate if the value is real, and its people are redeployed if it is not. Another year of funding because it has always existed is not on the list of options.

Leading a sunset

Retiring a product line that the teams built and love has to be led. That means giving the honest rationale, respecting what the work achieved, and laying a deliberate path for the people and for the users. The failure is to strangle it quietly by cutting its funding.

---

5. Scaling note

At team scope the manager is responsible for one team's roadmap and its platform debt, and holds the stated balance against pressure. At organisational scope the roadmap becomes a portfolio with admission criteria. Over-commitment is read from flow signals and corrected by cutting, chronic drains such as unplanned work are made visible and gated, and every standing investment has to state its mandate. At executive scope the question becomes the balance of horizons: how much capacity serves the current business, how much builds the platform the strategy needs, and how much explores. Long bets are staged, committed to before they start and defended from erosion, and existential priorities are funded by explicit subtraction. The pattern is in How Judgment Scales, and ranking the portfolio is covered in DE-1 Prioritisation under constraint.

---

6. Judgment

- Protects a stated share of capacity for foundational work and holds it under pressure. [team]
- Justifies platform investment in business terms. [team]
- Presents the roadmap as adaptable bets with confidence levels instead of dated promises. [team]
- Names what is deferred and what is not being done. [team]
- Failure mode: feature pressure consuming all capacity until velocity collapses. [team]
- Failure mode: gold-plating the platform. [team]
- Failure mode: roadmap as promise. [team]
- Failure mode: infrastructure justified purely on engineering grounds. [team]
-

Admits initiatives to the department roadmap by outcome contribution against cost and risk. [org]

- Reads over-commitment from flow signals and corrects it by cutting. [org]
- Makes chronic unplanned work visible, gates and owns intake, and either budgets explicit interrupt capacity or routes the rest into planning. [org]
- Requires every standing investment to state its mandate and success measures. [org]
- Leads a sunset with honesty and respect, and with a path for the people. [org]
- Failure mode: a roadmap that is the union of stakeholder requests. [org]
- Failure mode: treating chronic unplanned work as weather. [org]
- Failure mode: legacy teams funded by inertia. [org]
- Failure mode: sunsets communicated as an afterthought. [org]
- Owns an explicit mix of time horizons at executive level. [exec]
- Stages a long platform bet for intermediate value, sizes the near-term sacrifice honestly, and secures real commitment before starting. [exec]
- Makes the case for a long investment again, in current business terms, when short-term pressure threatens it; offers staged options; refuses the quiet nibbling of "temporary" reallocations. [exec]
- Sizes a bet on a disruptive technology to teach, with explicit checkpoints to double down or fold. [exec]
- Funds an existential new priority by explicitly clearing something out instead of adding it to everyone's plate (see TJ-5 AI-assisted engineering). [exec]
- Failure mode: endless near-term optimisation while the platform ages into a crisis. [exec]
- Failure mode: long bets launched without secured commitment. [exec]
- Failure mode: token innovation spend. [exec]
- Failure mode: adding the existential priority as everyone's tenth job. [exec]

7. Tensions

Now versus later. Every unit of capacity spent on the platform is a unit not spent on visible value this quarter. Every unit spent on features is borrowed from a future that will send the bill with interest. The stated balance resolves this. The judgment is in choosing the number, and in changing it explicitly instead of letting it erode.

Commitment versus adaptability. Stakeholders want the roadmap to be a promise, and reality makes it a forecast. Separating near-term commitments from directional bets, and labelling which is which, lets both be true.

Admission criteria versus relationships. A roadmap with criteria turns down requests from people the manager needs. A roadmap without them becomes the union of their wishes. When the criteria are applied visibly and explained in the requester's terms, a refusal is about the work and leaves the relationship intact.

Persistence versus sunk cost. A long bet needs protecting from every quarter's urgency, and a bad bet needs stopping whatever has been spent on it. What separates the two is whether the fundamentals have changed, along with the honesty to apply that test to a bet one has championed.

Focus versus exploration. Capacity spent exploring what might disrupt everything is capacity taken from what pays today, and an organisation that never explores gets disrupted by one that did. The mix of horizons should be decided, never left as whatever remains, and on both sides the failure is letting it happen by accident.

8. Worked scenario

A team's manager is in quarterly planning with their product partner. Over two quarters the team's velocity has fallen by a third. The cause is known. Every feature touches a data layer that nobody has had the capacity to fix, so each feature now costs more and breaks more. The product partner's roadmap for the coming quarter takes up the whole team, and they have said, reasonably from where they sit, that the business cannot afford any reduction in feature output this quarter.

One pull is to agree and hope: accept the full feature roadmap, promise to fit the data-layer work in "around the edges", and watch it fail to happen for a third quarter. The other is to declare a quarter of technical debt, which the business cannot afford and the partner will not accept.

The manager makes the case in the partner's terms. They show what the data layer costs now: cycle time up by a third, two incidents last quarter traced to it, and one roadmap feature that could not be built as specified because of it. They show the trajectory: at the current rate of decay, next quarter's roadmap will deliver less than this quarter's with the same headcount. Then they propose a stated balance in place of a pause. For the next two quarters, 25% of the team's capacity goes to the data layer. The work is staged so that the first six weeks deliver the change that unblocks the feature the partner wants most. The feature roadmap is worked out again to fit what the remaining 75% can deliver, and the partner chooses what moves.

The partner pushes back on the 25%. The manager keeps the number and changes the period instead: two quarters at 25%, then back to a standing 15%, reviewed quarterly. They also name what the team will not do. Two lower-priority features are deferred explicitly, so nothing is quietly dropped and the roadmap's non-goals are visible.

The partner agrees. The case was made in terms of delivered outcomes instead of engineering health, and the first stage delivers something the partner wanted. The manager did not accept the full roadmap and try to fit the platform work around it, because that decision produced the last two quarters.

9. Related competencies

- SV-1 Engineering strategy: the strategy for which the roadmap is the current sequence of bets.
- DE-1 Prioritisation under constraint: ranking the portfolio, and finishing fewer things.
- SV-4 Engineering economics: the economics of an investment case, and sunk cost in full.
- TJ-2 Technical debt: technical debt as the platform investment most often deferred.
- TJ-5 AI-assisted engineering: the existential AI priority, and funding it by subtraction.

10. Self-check

1. What happens to a team with zero capacity allocated to platform and debt? *Answer:* Velocity decays under accumulating debt and toil until it collapses. At that point the platform gets a crisis budget that costs more than a standing share would have. Zero is a decision, and it has a date on which it fails.
2. How is a platform investment argued to a product partner? *Answer:* In business terms. The case covers what the absence of the work costs now (cycle time, incidents, features that cannot be built), what the work enables, and the trajectory without it. Engineering elegance has no place in it.
3. Name three flow signals of an over-committed department roadmap, and what correcting them requires. [org] *Answer:* Everything in progress and nothing finishing; no capacity for the unplanned; dates missed in bulk. A fourth is no slack for discovery. Correcting them requires cutting work, and exhortation will not do it.
4. A long-running internal tools team cannot state its strategic value. What does it get? [org] *Answer:* An assessment against the strategy. If the value is real, it gets a clear mandate and success measures; if it is not, its people are redeployed. It does not get another year of funding because it has always existed.

5. Why does a two-year platform bet need staged value and explicit executive commitment before it starts? [exec] *Answer:* Without intermediate value, "temporary" reallocations made under each quarter's pressure will nibble it to death. Without secured commitment, it starts on borrowed patience that runs out before it delivers.
6. How is a disruptive-technology bet sized? [exec] *Answer:* To teach. It should be big enough to build real capability and learn fast, with explicit checkpoints to double down or fold. A bet too small to matter and one too big to survive being wrong are both errors.
7. An existential capability must be built and the organisation is full of committed work. What is the funding move? [exec] *Answer:* Explicit subtraction: clear something out to pay for focused bets and capability-building. Piling the priority onto everyone's full plate makes it everyone's tenth job and nobody's first.

Sources

- Mehrdad Baghai, Stephen Coley and David White, The Alchemy of Growth: Practical Insights for Building the Enduring Enterprise (1999): the three horizons of a portfolio.
- Hal R. Arkes and Catherine Blumer, The Psychology of Sunk Cost (1985): the psychology of sunk cost.

SV-4 Engineering economics

headcount, budget, cost of delivery, vendors, and build/buy/AI decisions

1. Definition and why it matters

Every allocation of engineering effort is a choice made against alternatives. v1.0 framed this competency as resource allocation and build-versus-buy. v3.0 renames it engineering economics because its scope has grown. It now covers the cost of delivery as a whole; budgets and headcount as investments that have to be argued for and defended; vendors and the dependencies they create; and a third option beside build and buy, the AI service, which has its own profile of cost and risk. The disciplines themselves are old. Build what differentiates and buy the commodity. Weigh total cost of ownership over sticker price. Concentrate effort so that work gets finished, instead of spreading it thin. Let what has already been spent argue for nothing. What has changed is that managers at every scope are now expected to reason about engineering in terms of cost and return, and the examinations reward the option that ties a decision to the economics over the one that follows engineering preference. Engineering is usually a company's largest discretionary cost, and a manager who cannot explain what it buys will be told what it costs.

2. Core principles

1. **Build the core, buy the context.** Build what differentiates the business and where control matters. Buy or adopt the commodity, which frees scarce engineers for differentiating work. Core means differentiating, which is not the same as important.
2. **Weigh total cost of ownership, not sticker price.** Building is never a one-time cost, because maintenance lasts for ever. Buying is not free either, once integration, lock-in and exit are counted. The discipline is to name the lifetime costs explicitly; computing them precisely is not the point.
3. **What has been spent argues for nothing.** Only the forward-looking comparison counts: value remaining against cost remaining, set against the alternatives. Changed fundamentals justify switching a bet. Novelty does not.
4. **Concentrate to finish.** With too many bets running at once, none of them is funded properly. Funding one decisively beats hedging into two mediocre efforts.
5. **Argue headcount and budget as investment, in outcomes.** Say what the spend enables and protects and what not spending costs, and stage it with checkpoints. A cut goes to the lowest-value work. It is never spread evenly as a way of avoiding a decision.

6. **Weigh control and dependency, not only cost.** For a strategic capability, two questions outrank price and speed: who ends up owning it, and where you stand if a vendor or partner relationship sours.

---

3. Models and evidence

The economic disciplines here are practitioner frameworks. The one bias with a research literature behind it is graded accordingly.

Core versus context [practice]

Geoffrey A. Moore, Dealing with Darwin: How Great Companies Innovate at Every Phase of Their Evolution (2005) sets out the distinction. Core is whatever differentiates a business in its customers' eyes. Context is everything else that has to be done without differentiating. Build and invest in core; buy, adopt or outsource context; and move engineers from context to core. The boundary shifts as yesterday's differentiator becomes a commodity. The usual misreading treats "core" as meaning "important". Email is important and is almost never core.

Total cost of ownership [practice]

This is the full lifetime cost of a capability. For a build it is the build cost plus maintenance for ever. For a purchase it is the price plus integration, lock-in and the cost of exit. It has no single author, and it corrects every claim that something is "cheaper". Costs over a long horizon are estimates. The discipline is to name them explicitly so they can be argued over, without computing them to a false precision.

Sunk-cost discipline [research]

Hal R. Arkes and Catherine Blumer, The Psychology of Sunk Cost (1985) found that prior investment makes people more likely to continue a course of action, regardless of its prospects. In engineering economics this governs three familiar cases: the half-done migration whose justification has worn away, the championed bet that a more promising direction has overtaken, and the legacy system kept because so much went into it.
The recommendation always looks forward: continue, descope or stop, judged on value remaining against cost remaining. Executive integrity means applying the same arithmetic to one's own advocacy.

Build, buy, or AI service [practice]

This is a decision, not a named model, and v3.0 adds it as a three-way choice: besides building a capability or buying a product, a team can adopt an AI service that provides it. The third option has a profile of its own. Its initial cost is low and its ongoing cost uncertain, the alternatives go out of date quickly, and it depends on a provider whose terms and capabilities change. It is weighed with the same questions as the other two. Is this core? What is the lifetime cost? Where do we stand if the provider changes course?

Headcount as investment [practice]

This is a framing, not a named model, and the unit treats it as essential for anyone who argues for a budget. Engineering headcount is an investment with a return. It is argued in terms of what it enables and protects and what its absence costs, staged with checkpoints, and defended with outcome measures in place of activity. The failure is a budget defended by describing how busy everyone is.

---

4. Practice

The build/buy/AI worksheet

Any capability decision larger than a sprint gets a one-page comparison. It asks whether the capability is core or context; sets out the lifetime cost of each option, including maintenance, integration, lock-in and exit; and assesses reversibility. For a strategic capability it also asks who ends up owning it and what happens if the relationship sours. The recommendation follows from the page, and the page is kept so the decision can be revisited when the boundary moves.

The forward-looking review

A bet that has consumed more than its budget, or whose justification has weakened, gets a review that ignores what has been spent. The review asks what value remains, what it will cost to obtain, and what the alternatives would return for the same spend. It ends in continue, descope or stop. The reviewer applies it with the same rigour to bets they championed.

The investment case

Headcount or budget is requested as an investment. The request sets out what it enables, what it protects, what not funding it would cost and how success will be measured, and it is staged so that the first tranche can be judged before the second is released. A cut is answered with the same structure. First identify the work that would not be started today, then aim the cut at it.

The vendor register

The register lists every significant vendor and AI service the department depends on. For each it records what the service does, what it costs, what leaving it would cost, and how much of the department's delivery would stop if it failed. It is reviewed every year. A dependency that could stop delivery and has no exit plan is a risk to be engineered down.

Charter clarity

Sometimes a peer department encroaches on a capability the team owns, or an opportunity with a high return turns up outside the team's charter. The right move is a conversation about the boundary at the level where charters are set, either to clarify the charter or to propose an explicit change. Duplication fought commit by commit is the wrong move, and so is a quiet land-grab. How charters are designed is covered in SV-6 Organisational design and leading change.

---

5. Scaling note

At team scope the manager deals with build-versus-buy calls and the team's own costs, and applies core versus context, total cost of ownership and focus directly. At organisational scope the work becomes a department budget and headcount plan. The manager chooses between bets of similar size on outcome, cost of delay, risk and reversibility; cuts under constraint by targeting the lowest-value work; and settles charter disputes at the level that owns them. At executive scope the manager is responsible for the economics of engineering as a whole (cost of delivery, capital allocation, headcount as investment) and for decisions with strategic consequences beyond engineering. These include build-versus-partner where the partner could become a competitor, funding one of two compelling bets properly instead of splitting the money to avoid choosing, and treating an acquisition's integration as a strategy with a bill attached. The pattern is in How Judgment Scales, and governance of the third option is covered in TJ-5 AI-assisted engineering.

---

6. Judgment

- Builds the differentiating core and buys the commodity context. [team]
- Weighs total cost of ownership over sticker price. [team]
- Concentrates effort to finish instead of spreading it across too many bets. [team]
- Revisits allocations on evidence as priorities shift. [team]
-

Evaluates an AI service as a third option with its own cost, obsolescence and dependency profile.

- Failure mode: building commodity capability because it was not invented here. [team]
- Failure mode: outsourcing the differentiator. [team]
- Failure mode: ignoring maintenance that lasts for ever. [team]
- Failure mode: too many simultaneous bets. [team]
- Compares bets of similar size explicitly on expected outcome, cost of delay, strategic fit, risk and reversibility, and not on whose advocacy is loudest. [org]
- Switches a committed bet only on changed fundamentals, never because something newer has appeared. [org]
- Reassesses a half-done, over-budget migration on forward-looking arithmetic, ending in a clean continue, descope or stop. [org]
- Targets a budget cut at the lowest-value work (what would not be started today), never at an across-the-board shave. [org]
- Settles a charter dispute explicitly at the level where charters are set. [org]
- Takes a high-return opportunity outside the team's charter to its owner, or proposes a charter change, instead of grabbing the territory or ignoring it. [org]
- Plans and defends a department budget and headcount in terms of outcomes and cost of delivery. [org]
- Failure mode: finishing doomed work to honour its sunk cost. [org]
- Failure mode: chasing novelty mid-bet. [org]
- Failure mode: peanut-butter budget cuts. [org]
- Failure mode: fighting charter wars through duplicated engineering. [org]
- Weighs a build-versus-partner decision for a strategic capability mainly on control and dependency risk: who owns the capability, and where you stand if the relationship sours. [exec]
- Funds one of two compelling bets properly, weighing the asymmetry (which failure is survivable, which success compounds) instead of splitting to avoid the decision. [exec]
- Applies the same forward-looking arithmetic to a bet they personally championed, and redirects when a better direction has emerged. [exec]
- Treats an acquisition's integration as an explicit strategy with a bill: which platform consolidates, which teams merge, on what timeline, with the cost of retaining the people the company bought priced in. [exec]
- Reasons about the economics of the whole function, such as cost of delivery and capital allocation, and changes them. [exec]
- Failure mode: partnering the crown jewels to a future competitor. [exec]

- Failure mode: splitting funding to avoid choosing. [exec]
- Failure mode: letting acquired stacks and organisations drift unintegrated. [exec]
- Failure mode: optimising acquisition cost while the acquired talent walks out. [exec]

7. Tensions

Control versus cost. Buying is usually cheaper, and building keeps the capability in hand. For context, buy; for core, build. The hard cases are capabilities sitting on a moving boundary, and the judgment is in seeing which way that boundary is moving.

Focus versus optionality. Funding one bet properly means giving up the other, and keeping both alive funds neither. Asymmetry decides it: which failure is survivable, which success compounds, and which bet builds capability the strategy needs anyway.

Persistence versus sunk cost. From the inside, the bet that needs one more quarter looks the same as the bet that should have been stopped a year ago. Only the forward-looking review tells them apart, and it is hardest to apply to one's own advocacy.

Speed versus dependency. An AI service or a partner delivers a capability now, and it also decides who owns that capability later. For context, speed wins. For a strategic capability, the dependency question outranks it.

Targeted cuts versus fairness. A cut aimed at the lowest-value work lands on particular teams and looks unfair. A cut spread evenly looks fair and degrades everything. The judgment is to make the targeting explicit and its criteria visible, so that the unfairness falls on the work and spares the people.

8. Worked scenario

A head of engineering has to decide how the company will get the AI infrastructure its products will increasingly depend on: the model serving, retrieval and evaluation layer that the next two years of product strategy assume. Three options are on the table. The first is to build it, which the platform team estimates at a year with four senior engineers. The second is to buy it from a vendor whose product is good, whose pricing is per request and whose roadmap is opaque. The third is a partnership with a well-funded company in an adjacent market. It has offered to provide the layer cheaply in exchange for close integration, and, on the evidence of its own strategy, it could become a competitor within two years.

The pull is toward the partner. It is the fastest and cheapest option, the offer is already on the table, and the chief financial officer favours it. The head of engineering starts instead with the question that outranks cost and speed for a strategic capability: is this core, and who ends up owning it? The layer is core, since it is what the product strategy differentiates

on. Under the partnership it would be owned by a company with every incentive to withdraw it at the moment it mattered most. The company's position if the relationship soured carries the most weight, and that position would be untenable.

Next the vendor option is weighed on total cost of ownership instead of sticker price. That means per-request pricing against projected volume, the cost of integration, the cost of leaving if the vendor's roadmap diverges, and the dependency on a provider whose terms can change. The build option is weighed in the same terms: a year of four senior engineers, maintenance for ever, and the opportunity cost of what those engineers would otherwise do. Against that, the company would own a core capability and could differentiate on it.

The recommendation is staged. The commodity parts of the layer are bought from the vendor now, with an exit plan and a contract that caps the pricing risk. The differentiating parts, retrieval and evaluation, where the product's edge lies, are built over the same year by two engineers instead of four. The partnership is declined, and the head of engineering gives the chief financial officer the reasoning in the company's terms: the offer is cheap because the partner would own the thing the strategy depends on.

Deciding on cost and speed would have chosen the partner and handed a future competitor the capability that mattered. The head of engineering did not make that decision.

9. Related competencies

- SV-3 Roadmapping: the roadmap's platform investment as the recurring allocation decision.
- SV-2 Communicating and partnering across functions: arguing the investment case to executives and the board.
- SV-6 Organisational design and leading change: charters and ownership boundaries as the structural form of allocation.
- TJ-5 AI-assisted engineering: governing the AI option, building capability for it, and its risk.
- DE-1 Prioritisation under constraint: the portfolio ranking that allocation decisions serve.

10. Self-check

1. How does core versus context guide a build-versus-buy decision, and what is the common misreading? *Answer:* Build what differentiates the business and where control matters, and buy the commodity. The misreading is to treat "core" as meaning "important". Email is important and almost never core.

2. What does total cost of ownership add to a "this is cheaper" argument? *Answer:* The lifetime costs that the sticker price hides: maintenance for ever for a build, and integration, lock-in and exit for a purchase. The discipline is to name them explicitly; precise computation is not the point.
3. A year-old migration is half done, over budget, and its justification has weakened. What does the recommendation weigh, and what does it ignore? [org] *Answer:* It weighs value remaining against cost remaining, compared with the alternatives, and ends in continue, descope or stop. It ignores what has already been spent, which argues for nothing.
4. You are told to cut the department budget by 15%. Where do you look first, and what is the wrong answer? [org] *Answer:* At the work with the lowest strategic value, meaning the initiatives that would not be started today. The wrong answer is the across-the-board shave, which degrades everything equally and decides nothing.
5. What is different about the AI service as a third option beside build and buy? *Answer:* Low initial cost, uncertain ongoing cost, alternatives that go out of date quickly, and dependency on a provider whose terms and capabilities change. It is weighed with the same questions as the others: core or context, lifetime cost, and where you stand if the provider changes course.
6. In a build-versus-partner call where the partner could become a competitor, what weighs most and why? [exec] *Answer:* Strategic control and dependency risk: who ends up owning the capability, and where you stand if the relationship sours. For a core capability these outrank cost and speed, because the offer is cheap precisely because the partner would own what the strategy depends on.
7. What makes redirecting from a bet you personally championed the credibility move rather than the credibility loss? [exec] *Answer:* Applying the same forward-looking arithmetic to your own advocacy is what makes your next recommendation believable. Doubling down to protect the appointment or the argument loses the bet and the credibility together.

Sources

- Geoffrey A. Moore, Dealing with Darwin: How Great Companies Innovate at Every Phase of Their Evolution (2005): core versus context.
- Hal R. Arkes and Catherine Blumer, The Psychology of Sunk Cost (1985): the psychology of sunk cost.

SV-5 Engineering metrics and measurement

choosing, interpreting, and not gaming them, including measuring AI impact

1. Definition and why it matters

Metrics make engineering legible and improvable, and the wrong metrics do active harm. v1.0 framed the competency as setting and using meaningful measures. v3.0 adds interpretation and the measurement of AI impact, because the pressure to show a number is now strongest exactly where the numbers are least settled. The competency starts with choosing a small, balanced set of measures that describe the delivery system and leave the people in it alone. Those measures are read as trends that open a conversation, never as verdicts. The manager refuses the uses that corrupt them: individual productivity scores, leaderboards across teams, a single number for the board. At executive scope the competency also means declaring in advance how the organisation will know whether its strategy is working. Every metric becomes a target as soon as it is tied to a reward or a comparison, and growth that is not measured cannot be told apart from bloat. The examinations test this through the productivity score per engineer, the league table of teams, the metric that improved while nothing else did, and the AI investment that has to show a return the honest number does not yet support.

2. Core principles

1. **Measure the system, not the people.** The validated delivery measures describe the team's delivery system, its throughput and stability. They are never individual productivity scores.
2. **A measure that becomes a target stops measuring.** Activity metrics such as lines of code, commit counts and story points per engineer are the easiest to game and do the most damage when turned into weapons. Work out how a measure will be gamed before attaching a target to it.
3. **Small, balanced, and paired with context.** Keep a few measures that span throughput and stability, so that improving one cannot quietly wreck another, and put the numbers next to the qualitative signals that explain them.
4. **Trend a team against itself; never rank teams against each other.** Contexts differ, story points are not a currency, and a ranking teaches every team to optimise the number.
5. **Instrument the system between teams.** Above team scope, the thing only the manager owns is the set of queues, handoffs and dependencies between teams.

Measure those across the whole group instead of measuring each team's internals one by one.

6. **Declare the success measure when the strategy is set.** A strategy with no falsifiable measure cannot be shown to be failing, and its "success" will be judged by the effort spent on it. Report AI's effect on the same terms: honestly, with the uncertainty stated, including where the effect is negative.

---

3. Models and evidence

No area of this standard has more research behind it than measurement, or more misuse of that research, and the unit grades both carefully.

DORA metrics [research]

The research programme in Nicole Forsgren, Jez Humble and Gene Kim, Accelerate: The Science of Lean Software and DevOps (2018) validated four delivery measures. Deployment frequency and lead time for changes measure throughput; change failure rate and time to restore service measure stability. It also found that in high-performing organisations the two move together and do not trade off. They are deliberately team-level measures of the delivery system, meant for trending a team against its own history. Used to measure individuals, or turned into a leaderboard across teams, they become targets and the research no longer applies.

The SPACE framework [practice]

Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck and Jenna Butler, The SPACE of Developer Productivity: There's More to It Than You Think (2021), written by several of the same researchers, argues that no single metric can capture developer productivity. It proposes five dimensions to be measured in balance: satisfaction and wellbeing, performance, activity, communication and collaboration, and efficiency and flow. In practice it works as a checklist against one-dimensional measurement. Any proposed productivity metric that sits in only one dimension, and above all one that sits in activity alone, is incomplete by construction. The paper proposes a framework and reports no empirical finding, and it is graded on that basis.

Goodhart’s law [practice]

In the form given by Marilyn Strathern, 'Improving Ratings': Audit in the British University System (1997): when a measure becomes a target, it ceases to be a good measure. This is the hazard governing everything in the unit, at every scope. It shows up as the engineer who games a personal metric, the team that optimises a leaderboard, and the organisation

whose activity dashboard looks healthy while outcomes stay flat. The law does not say stop measuring. It argues for balanced sets, for measures anchored in outcomes, and for treating metrics as the start of a conversation instead of a verdict.

Measuring AI impact [contested]

DE-5 Designing how work flows introduces the evidence. In a controlled experiment, Sida Peng, Eirini Kalliamvakou, Peter Cihon and Mert Demirer, The Impact of AI on Developer Productivity: Evidence from GitHub Copilot (2023) found that a defined task was completed substantially faster with an AI assistant. In a randomised trial, Joel Becker, Nate Rush, Elizabeth Barnes and David Rein (METR), Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity (2025) found experienced developers working slower in their own repositories while believing they were faster. The lesson for measurement is specific. Self-reported speed-up is not a measure. Adoption and licence counts are not measures either, and pull-request counts measure activity. The effect has to be measured in the organisation's own conditions, against outcomes, with the uncertainty stated. The honest report to a board may be "not yet, and here is why".

The falsifiable scorecard [practice]

This is a practice, not a named model, and the unit treats it as the executive's answer to "is the strategy working?". It is a small, stable scorecard declared when the strategy is set. It covers the business outcomes the investment was meant to move, leading indicators chosen in advance, trends in the delivery system at organisation level, and organisational health. It is reviewed on a cadence, revised whenever the strategy is, and able to prove the strategy wrong. It fails in two ways: the dashboard swells until nobody reads it, or a board meeting discovers that "working" was never defined.

---

4. Practice

The team’s four numbers

Deployment frequency, lead time, change failure rate and time to restore are trended monthly against the team's own history and read in the retrospective alongside the team's account of what happened. The numbers find the problems, and the conversation explains them. Nobody's name appears next to any of them.

The gaming pre-mortem

Before tying any metric to a target, a reward or a comparison, the manager asks how a reasonable person could hit the number without doing the thing it stands for. If that would

be easy, the target is withdrawn, or the measure is paired with another that would expose the gaming.

The between-teams instruments

A group of teams needs measures of the system that connects them: queue times at shared dependencies, handoff latency, and how long cross-team requests wait. These instruments belong to the manager of managers. They show capacity drains and platform decay before delivery does.

Measuring AI in the team's conditions

A rollout of AI assistance is measured against outcomes such as cycle time, change failure rate and rework, in the organisation's own codebases. The period has to be long enough to show effects on understanding as well as on speed. Self-reports are collected, but they are not the measure. The report states the uncertainty.

The scorecard, declared in advance

When a strategy or a major investment is set, the executive writes down how the organisation will know whether it is working. That means the outcomes, the leading indicators, the trends in delivery and organisational health, and the date of the first review. The scorecard fits on one page, and that page is what the board sees.

5. Scaling note

At team scope the manager looks after one team's metrics and what they hide. They measure the delivery system, pair throughput with stability and refuse to score individuals. At organisational scope a new temptation appears: comparing teams. The manager's task becomes finding measures that aggregate or compare without inviting the gaming that comparison creates. Teams are trended against themselves, the system between them is instrumented, and measured evidence goes into every argument about capacity. At executive scope the question is whether the strategy is working. The executive chooses the measures the business uses to judge engineering, picking ones the organisation can hit without being made worse; declares a falsifiable scorecard in advance; and reports AI's effect honestly to the board. The pattern is in How Judgment Scales, and the flow the measures describe is covered in DE-5 Designing how work flows.

6. Judgment

- Measures delivery outcomes at team level and pairs throughput with stability. [team]

STRATEGY & VISION

- Uses metrics to find problems, never to rank people. [team]
- Anticipates gaming before attaching a target, and stays alert to it afterward.
- Declines to use a single productivity number for individuals, and can say why.
- Failure mode: individual output treated as "productivity". [team]
- Failure mode: chasing one metric until another breaks. [team]
- Failure mode: metrics as surveillance. [team]
- Failure mode: numbers without context. [team]
- Trends each team against itself (cycle time and its direction, predictability, change failure rate) and never puts teams on a leaderboard. [org]
- Instruments the system between teams: queues, handoffs, dependencies. [org]
- Brings measured evidence to every argument about capacity, drains and platform decay. [org]
- Measures the effect of AI assistance honestly across teams, in their own conditions, including where it is not helping. [org]
- Failure mode: velocity leaderboards across teams. [org]
- Failure mode: normalising story points into a fake currency. [org]
- Failure mode: measuring team internals while the queues between teams go dark. [org]
- Failure mode: an AI rollout where the only number that moved is pull-request count. [org]
- Declares the strategy's success criteria when the strategy is set, and keeps the scorecard small and stable. [exec]
- Chooses the measures by which the business judges engineering, and refuses any measure the organisation would be made worse by hitting. [exec]
- Reports AI's effect to the board with its uncertainty, including a negative or not-yet result. [exec]
- Treats growth without measured outcomes as indistinguishable from bloat, and fixes the measurement instead of the narrative. [exec]
- Failure mode: declaring victory on effort expended. [exec]
- Failure mode: a strategy with no falsifiable success measure. [exec]
- Failure mode: dashboard sprawl. [exec]
- Failure mode: a single productivity number for the board. [exec]

7. Tensions

Legibility versus distortion. Every measure that makes engineering legible to the business also invites the organisation to optimise it. A balanced set and the gaming pre-mortem deal with this. Giving up measurement does not.

Comparison versus context. Leadership wants to compare teams, and comparison is what turns a measure into a target. Trending each team against itself tells leadership what it needs to know, whether this team is improving, without a leaderboard.

Evidence versus timing. Measuring an AI investment honestly takes longer than the quarter in which the board wants to see a return. The judgment is to report the uncertainty instead of the convenient number, and to set the review date at the moment the investment is made.

Simplicity versus completeness. A single number is what executives ask for and what corrupts fastest. A full dashboard is complete and goes unread. The one-page scorecard, declared in advance, is the compromise that survives both pressures.

Measuring versus trusting. A manager who instruments everything signals distrust, and one who instruments nothing manages by anecdote. The line falls between the system and the people. Measure the delivery system thoroughly and never measure the individual.

8. Worked scenario

A senior manager leads six teams. Their vice-president asks for a monthly league table ranking the teams by velocity, so that leadership can see which teams are performing. Two of the teams work on a mature product with heavy compliance requirements. Two work on a new product with none. Two are platform teams whose output the other four consume. The vice-president's aim is reasonable, since they cannot tell which teams are in trouble, and the request is specific.

Producing the table is the easy answer, and it would damage all six teams. Story points are not a currency that works across teams. The contexts differ so much that the ranking would measure the product instead of the team. Within a month every team would learn to optimise the number, which is the one thing leadership does not want. Refusing the request is the other wrong answer. It leaves the vice-president unable to see what they need to see, and it looks like engineering resisting accountability.

The senior manager answers the need behind the request. Within a month they propose and build a view that gives leadership what it wants. Each team is trended against itself on cycle time, predictability (committed against delivered) and change failure rate. A team whose cycle time is rising or whose predictability is falling becomes visible without being ranked against a team in a different context. Next to this sit measures of the system between the teams: how long requests to the platform teams wait, and how often cross-team dependencies slip. The trouble the vice-president suspects usually lives there, and that is the part of the system the senior manager alone owns.

They also show the vice-president, using two months of history, what the league table would have done. By velocity the "worst" team was the compliance team, whose cycle time was flat and whose change failure rate was the lowest of the six. The "best" was a new-product team whose predictability had collapsed. The table would have rewarded the team in trouble and punished the one performing well.

The vice-president accepts the trended view. The senior manager neither produced a ranking for the sake of appearing to comply nor refused and left the need unmet.

9. Related competencies

- DE-5 Designing how work flows: the flow the measures describe, and Goodhart's law in the operating cadence.
- DE-3 Reliability, incident response, and operational ownership: the stability measures, and the research showing throughput and stability move together.
- SV-1 Engineering strategy: the strategy whose success measure is declared when it is set.
- SV-2 Communicating and partnering across functions: bringing measured evidence into the conversation with the board.
- TJ-5 AI-assisted engineering: the AI adoption whose effect is being measured.

10. Self-check

1. Why are the DORA metrics team-level rather than individual? *Answer:* Because they describe the delivery system's throughput and stability, not a person's output, and the research that validates them applies to that use. Used as individual measures they become targets and stop measuring.
2. What question is asked before any metric is attached to a target? *Answer:* How would a reasonable person hit this number without doing the thing it stands for? If that would be easy, withdraw the target or pair the measure with one that would expose the gaming.

3. Why is trending a team against itself sound where ranking teams against each other is not? [org] *Answer:* A trend answers "is this team improving?" within the team's own context. A ranking compares contexts that differ, treats story points as a currency, and teaches every team to optimise the number.
4. What does a manager of managers uniquely own, and therefore measure? [org] *Answer:* The system between teams: queues at shared dependencies, handoff latency, and cross-team requests waiting. That is where capacity drains and platform decay show up before delivery suffers.
5. An AI-tooling rollout must show a return and the only number that moved is pull-request count. What is the honest position? [org] *Answer:* Pull-request count measures activity, not outcome. Measure against cycle time, change failure rate and rework in the organisation's own conditions, over a period long enough to see effects on understanding. Collect self-reports without treating them as the measure, and report the uncertainty.
6. What does it mean for a strategy to have a falsifiable success measure, and what happens at the board when it does not? [exec] *Answer:* It means outcomes and leading indicators declared when the strategy is set, able to show that the strategy is failing. Without them, "working" is judged by effort expended, and the board finds out that success was never defined. At that point growth cannot be told apart from bloat.
7. The board wants a single number for engineering productivity. What is the answer? [exec] *Answer:* A one-page scorecard declared in advance, covering strategic outcomes, delivery-system trends and organisational health, in place of a single number. Give the reason: any single number would make the organisation worse to hit.

Sources

- Nicole Forsgren, Jez Humble and Gene Kim, Accelerate: The Science of Lean Software and DevOps (2018): the DORA delivery metrics and the research behind them.
- Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck and Jenna Butler, The SPACE of Developer Productivity: There's More to It Than You Think (2021): the SPACE framework; productivity is not one number.
- Marilyn Strathern, 'Improving Ratings': Audit in the British University System (1997): Goodhart's law in its standard form.
- Sida Peng, Eirini Kalliamvakou, Peter Cihon and Mert Demirer, The Impact of AI on Developer Productivity: Evidence from GitHub Copilot (2023): the controlled experiment on AI-assisted task completion.

Joel Becker, Nate Rush, Elizabeth Barnes and David Rein (METR), Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity (2025): the randomised trial on experienced developers with AI tools.

SV-6 Organisational design and leading change

team structures, ownership boundaries, reorgs, and transitions

1. Definition and why it matters

Organisational design decides who owns what: the team boundaries, the charters, the layers between leaders and the work, and the decision rights attached to each. Leading change is what happens when that design changes, through a reorganisation, a merger, an acquisition or a change of leadership. Most designs go wrong at this stage, because a structure that is right on paper is only as good as the way it is introduced. v2.0 treated ownership through structure as the executive form of delegation. v3.0 makes it a competency in its own right, within Strategy & Vision, because the organisation is how the strategy gets implemented; announce a new strategy over the old structure and the old structure remains the real strategist. A reorganisation is the most consequential and least reversible decision most engineering leaders make. Chronic delivery friction at the same boundary is usually the structure presenting its bill. And repeated reorganisations that fail to stick teach an organisation that nothing will. The examinations assess this competency from EMP-II upward. It has no team-scope form.

2. Core principles

1. **Ownership is designed, not assigned conversation by conversation.** Structure, charters and decision rights determine who owns what. A reorganisation is above all a redesign of ownership, and it is judged by how clear and lasting the ownership it produces turns out to be.
2. **A team boundary and a system boundary are one decision.** Systems mirror the organisations that build them. Design the teams and the architecture together, or whichever one you left undesigned will reshape the other.
3. **Design around cognitive load and ownership, not around the chart's tidiness.** A team's scope is limited by what it can hold. A platform exists to reduce the load on the teams that ship. Spans of control depend on how mature the managers are and how stable the work is.
4. **Separate structural problems from people problems, in both directions.** Chronic friction at a boundary that persists after process fixes is structural. One struggling manager is a people problem. Mistaking either for the other wastes a year.
5.

Change is led, or it is suffered. Leading it means explaining why, saying what changes and what does not, attending to every leader whose scope moves, and staying present through the transition. An announcement on its own leads nothing.

6. **The change has to stick.** Reorganisation after reorganisation, each reversed or diluted, teaches cynicism, and cynicism costs more to repair than any other property of an organisation. Design the structure you intend to keep, then keep it.

---

3. Models and evidence

Organisational design has an old research literature and a recent practitioner one. The unit draws on both and grades each.

Conway's law [practice]

Melvin E. Conway, How Do Committees Invent? (1968) observed that a system's structure mirrors the communication structure of the organisation that built it. This unit uses the law as a design principle, not as a diagnostic. The boundaries drawn between teams will become boundaries in the system, so they should be drawn deliberately and at the same time. The practical form is the reverse manoeuvre, in which teams are shaped to produce the architecture you want. The law says the two structures correlate. It does not say which one to change to fix the other, and that choice is the manager's judgment.

Team topologies and cognitive load [practice]

Matthew Skelton and Manuel Pais, Team Topologies: Organizing Business and Technology Teams for Fast Flow (2019) provides a design vocabulary of four team types (stream-aligned, platform, enabling and complicated-subsystem) and three interaction modes, with the team's cognitive load as the constraint the design has to respect. Its value here is that it gives a vocabulary to the question of which teams should exist and how they should relate, so it no longer has to be settled by argument. It also states the platform team's purpose plainly: to reduce the load on the teams that ship product.

Sizing teams and spans [practice]

Will Larson, An Elegant Puzzle: Systems of Engineering Management (2019) takes a systems view of how large a team should be and how many teams one manager can lead, and argues that structure should follow from those constraints. Span of control is no constant. It depends on how mature the managers are, how stable the work is and how much coordination each report needs. A span that works in a steady state breaks during change, which is exactly when reorganisations happen.

Structure follows strategy [practice]

Alfred D. Chandler Jr., Strategy and Structure: Chapters in the History of the American Industrial Enterprise (1962) drew its thesis from a historical study of large firms. Changes in strategy come first and require changes in structure, and a structure that outlives the strategy that produced it becomes the obstacle to the next one. Here it supports the principle that the organisation is the strategy's implementation, along with its corollary: a strategy announced with no change in structure has probably not been implemented.

Leading change [practice]

John P. Kotter, Leading Change (1996) sets out an eight-step model: urgency, a guiding coalition, a vision, communication, removing obstacles, short-term wins, consolidation, and anchoring. This unit offers it as a checklist of what reorganisations leave out, without prescribing it as a procedure. Engineering reorganisations most often skip the last two steps, which are consolidating the change so that it is not quietly reversed and anchoring it in how decisions are in practice made. The model is prescriptive and its evidence comes from cases, so it is graded as practice.

---

4. Practice

The ownership map, before and after

Before any structural change, map every system, capability and decision against its current owner. Once the design is done, draw the same map with the new owners. Anything left without an owner in the second map is where the next incident will come from. The map gets published. The chart does not.

The design questions

Any proposed structure has to answer these questions. What does each team own, and what is its cognitive load? Which boundaries in the system will it create or remove? Who will have to coordinate with whom as a result? What is each manager's span? Which decisions move, and to where? A design that cannot answer them is only a chart.

The leader-impact review

Before the announcement, the review covers every leader whose scope changes. What do they gain and what do they lose? What will they be told, and by whom? What will the organisation need from them during the transition? A reorganisation that surprises its own leaders has nobody leading it.

The change narrative

A single written page covers why the change is happening, what changes, what explicitly does not, what people can expect in the first month, and what the leadership will visibly do differently. The leader delivers it in person, repeats it, and makes it part of the first decisions taken under the new structure. The check is whether a team lead can explain the reason a month later.

Reading the boundary

When delivery friction keeps recurring at the same boundary despite process fixes, the question to ask is structural. Are these two teams that should be one, or one that should be two? Is there a gap in ownership? Is a system boundary in the wrong place? A working group is the answer only if that question has been asked and the answer was no.

5. Scaling note

This competency has no team-scope form. A first-line manager works inside a structure someone else designed, and the EMA-I outline lists SV-6 as not assessed. At organisational scope the manager deals with team boundaries and ownership within a department, such as reorganising several teams, settling a charter dispute at the level that sets charters, or defining what a platform team is for, and leads those teams through the change. At executive scope the subject is the shape of the organisation itself: aligning structure with strategy; transitions, mergers and acquisitions; the layers between leaders and the work; and the cynicism that repeated change breeds. The general pattern is in How Judgment Scales. Delegating within the structure is covered in PL-1 Delegation, ownership, and decision rights, and the managers who sit in it in PL-6 Leading through leaders.

6. Judgment

- Designs team boundaries around ownership and cognitive load, and treats a team boundary and a system boundary as one decision. [org]
- Names what a platform team is for, and holds it to reducing the load on the teams that ship. [org]
- Leads several teams through a reorganisation, explaining why, what changes and what does not, and staying present through the transition. [org]
- Settles a charter dispute between teams explicitly, at the level that sets charters. [org]
- Recognises when a structural problem is being treated as a people problem, and the reverse. [org]

STRATEGY & VISION

- Reads chronic friction at the same boundary as a structural signal, and asks the structural question before adding a working group. [org]
- Checks the ownership map after any change and closes the gaps before the next incident finds them. [org]
- Failure mode: reorganising to tidy the chart instead of to clarify ownership. [org]
- Failure mode: announcing the change and calling it leadership. [org]
- Failure mode: patching with process a boundary the structure created. [org]
- Failure mode: a platform team with no stated purpose, or one that has become a bottleneck for everyone it was meant to unblock. [org]
- Failure mode: treating a rivalry between two teams as a personality problem when their charters overlap. [org]
- Aligns structure, skills and architecture with a new strategy all together, instead of announcing the strategy over the old chart. [exec]
- Weighs reorganisation options by their effect on leaders' scopes, ownership clarity and coordination cost, and not by who is asking. [exec]
- Leads a merger or an acquisition as a deliberate integration (which platform consolidates, which teams merge, on what timeline), with the cost of retaining the people the company bought priced in. [exec]
- Designs the layers between leaders and the work so that decisions are made at the level that has the information. [exec]
- Makes a change stick by consolidating it, anchoring it in how decisions are made, and holding it against pressure to reverse. [exec]
- Repairs the cynicism left by repeated reorganisations by making the next change smaller, explained, and held. [exec]
- Failure mode: new strategy, old organisation chart. [exec]
- Failure mode: repeated reorganisations, each reversed or diluted, until nobody believes change will stick. [exec]
- Failure mode: "let both stacks live" after an acquisition, which is a decision in itself and usually the most expensive one. [exec]
- Failure mode: reorganising without weighing the impact on the leaders whose scopes change. [exec]
- Failure mode: confusing empowered leaders with nobody being accountable. [exec]

7. Tensions

Clarity versus flexibility. Sharp ownership boundaries make accountability clear and make anything that crosses them slow. The judgment is in placing the boundaries to suit how the work in fact flows, and in designing how teams interact across them instead of pretending they will not need to.

Structure versus people. The structure that is right on paper may shrink a strong leader's scope or enlarge a weak one's. Designing only around the people freezes the structure to today's roster. Designing only around the work throws away what the people are good at. Both are weighed, and the weighing is written down.

Stability versus adaptation. An organisation that never changes its structure gets outgrown by its strategy, and one that changes every year cannot execute any strategy. Change has a real cost, paid in cynicism. So the structure is designed to be kept, and changed rarely and well.

Speed versus consent. A reorganisation decided quickly and announced without warning surprises the leaders it depends on. One negotiated with everyone it affects ends up designed by committee and satisfies nobody. The answer is to consult the leaders whose scopes move on the design questions, and to leave the decision with the person accountable for the outcome.

Integration versus disruption. After an acquisition, integrating fast destroys what was bought, and integrating slowly lets two stacks and two cultures harden. The judgment is a deliberate sequence with an explicit bill, so that neither direction is reached by drift.

8. Worked scenario

A senior manager runs five teams, and two of them are locked in a rivalry that has started to damage collaboration. They block each other's changes and escalate disputes they should settle between themselves. Each team's manager, separately, has described the other team as the problem. Six months ago the senior manager's first instinct was to treat this as a people problem. They arranged a facilitated conversation between the two managers and a shared retrospective, and set an explicit expectation that the two would cooperate. It helped for a month.

Looking again, the senior manager asks the structural question they skipped the first time. The two teams' charters overlap. Both own parts of the same customer-facing capability, divided along a system boundary drawn three years ago when the capability was smaller. Every significant change touches both teams' code, every release needs both of them to coordinate, and each team is rewarded for shipping its own part regardless of the other. What looks like a clash of personalities is two teams asked to share ownership of one thing, behaving exactly as the structure rewards them for behaving.

The senior manager writes down the design options. The first is to merge the two teams into one that owns the whole capability. The second is to split them again along a different boundary that matches how the capability is changed in practice. The third is to keep the structure and impose a shared cadence, which is the working-group approach already tried. A merger would produce a team of fourteen, more than one manager can serve well. Splitting along the boundary that matches the work gives each team something it owns entirely: one team takes the capability end to end, and the other takes a separate area that has lacked a clear owner.

The senior manager consults both managers on the design questions. One of them will lose scope, so that conversation comes first. It happens in private, names what they gain as well as what they lose, and the senior manager stays present through the transition instead of disappearing after the announcement. Both teams hear the reason for the change, which is that the structure was rewarding the rivalry, along with what explicitly stays the same. The ownership map is published again, and over the following quarter the system boundary is moved to match the team boundary, because the two are one decision.

A third facilitated conversation would have missed the point. The people were behaving as the design rewarded them for behaving, so the design was what had to change.

9. Related competencies

- PL-1 Delegation, ownership, and decision rights: ownership through structure as the executive form of delegation, and the hub-and-spoke bottleneck.
- PL-6 Leading through leaders: the managers who sit in the structure, and span of control as a design constraint.
- SV-1 Engineering strategy: the strategy the organisation implements, since structure follows strategy.
- DE-4 Managing scope, risk, and dependencies across workstreams: chronic friction at a boundary as the delivery symptom of a structural cause.
- TJ-1 Evaluating architectural decisions and their organisational consequences-: architecture and organisation as two views of one decision.
- CC-4 Motivation, recognition, and sustainable pace: leading distributed teams through change, and motivation and pace during a transition.

10. Self-check

1. Why is a team boundary and a system boundary one decision? [org] *Answer:* Because systems mirror the organisations that build them. The boundaries drawn between

teams become the boundaries in the system, so both are drawn together and on purpose. Otherwise the one left undesigned reshapes the other.

2. Two teams are in a rivalry that is hurting collaboration. What question is asked before the facilitated conversation? [org] *Answer:* Whether the structure is rewarding the rivalry. Do their charters overlap? Do they share ownership of one thing? Does every change need both of them? If so, the problem is one of design, and the people are behaving as the design rewards.
3. What is a platform team for, and what is the sign that it has stopped doing it? [org] *Answer:* Reducing the cognitive load on the teams that ship product. The sign that it has failed is that it has become the bottleneck for everyone it was meant to unblock.
4. What do reorganisations most often leave out, and what does leaving it out cost? [org] *Answer:* Leading the change. That means explaining why, saying what does not change, attending to each leader whose scope moves, staying present through the transition, and consolidating the change so it sticks. Without it the result is a chart nobody believes and a change that is quietly reversed.
5. Why is announcing a new strategy over the old organisation chart a failure of implementation? [exec] *Answer:* Because structure follows strategy. The organisation is how the strategy is implemented, and a structure left unchanged becomes the obstacle to a strategy it was not built for.
6. After an acquisition brings a redundant engineering organisation with its own stack, what is "let both stacks live"? [exec] *Answer:* A decision, and usually the most expensive one. Integration is led as an explicit strategy with a bill: which platform consolidates, which teams merge, on what timeline, with the cost of retaining the people the company bought priced in.
7. Repeated reorganisations have left an organisation cynical that change will stick. What repairs it? [exec] *Answer:* A next change that is smaller, explained, consolidated, and held against the pressure to reverse it. Only a change that sticks repairs cynicism, and cynicism is the most expensive property of an organisation to rebuild.
8. How are two reorganisation options weighed? [exec] *Answer:* By their effect on leaders' scopes, by how clear and lasting the ownership each produces is, and by the coordination cost each creates. Who is asking for which option does not come into it.

Sources

- Melvin E. Conway, How Do Committees Invent? (1968): systems mirror the organisations that build them.
- Matthew Skelton and Manuel Pais, Team Topologies: Organizing Business and Technology Teams for Fast Flow (2019): team types, interaction modes, cognitive load.

STRATEGY & VISION

- Will Larson, An Elegant Puzzle: Systems of Engineering Management (2019): sizing teams and spans; structure from constraints.
- Alfred D. Chandler Jr., Strategy and Structure: Chapters in the History of the American Industrial Enterprise (1962): structure follows strategy.
- John P. Kotter, Leading Change (1996): the steps of leading change, and the ones most often skipped.

Technical Judgment

What this domain is

Technical Judgment is the ability to exercise sound engineering judgment without doing the engineering yourself, and to own the technical responsibilities that cannot be delegated away. It is misunderstood at every scope, more than any other domain. New managers fear that stepping back from the code makes them irrelevant. New executives fear the opposite, that technical matters are now beneath them and can be handed off entirely. Both are wrong. Credibility comes from reasoning well about technical reality, not from writing the most code. Architecture, reliability, security and technical risk stay strategic concerns at the very top. The domain carries 20% of every examination: 12 questions on EMA-I, 15 on EMP-II and 18 on EME-III. Its six competencies are listed below with the questions each carries at each level, and the Exam Content Outlines give the proficiency indicators.

The throughline

Each competency in this domain puts judgment ahead of throughput. That means knowing enough to ask the right questions, telling real risk from hypothetical risk, deciding when good enough is good enough and, hardest of all, knowing which decisions are yours. v3.0 adds two responsibilities that v1.0 left implicit and the job no longer can. One is AI-assisted engineering, where the evidence is mixed and the gap between a considered judgment and a fashionable one is widest. The other is security, privacy and compliance, which engineering owns instead of delegating to a separate function. The examinations favour the option that asks the sharp question, demands the evidence, installs the decision process, or takes the risk to the level that should decide it. They mark down the option that rubber-stamps, rewrites, overrides or defers.

How the domain scales

At team scope you exercise judgment about one team's technical decisions, mostly through questions and restraint. At organisational scope you keep architecture and technical health coherent across teams while staying credible at one remove. That takes standards at the boundaries, deliberate platform bets, class-level fixes with named owners, and a decision process installed before any opinion is issued. At executive scope you own the organisation's technical strategy: the bet-the-company decisions, the platform's runway treated as business risk, the strategy for AI-assisted engineering, the organisation's security exposure, and a short, explicit docket of the decisions that are yours alone.

The general pattern is in How Judgment Scales; each unit's scaling note places its own competency on it.

Reading order

Start with TJ-4 Technical credibility and intervention. It holds the domain's two governing ideas, systems understanding over currency and the intervention threshold, and every other unit applies them. Then read TJ-1 Evaluating architectural decisions and their organisational consequences and TJ-2 Technical debt together, since one covers the decisions that shape systems and the other their accumulated cost. TJ-3 Code review and quality standards is the quality system and introduces the standard for AI-generated code that TJ-5 AI-assisted engineering governs, so read them in that order. TJ-6 Security, privacy, and compliance as engineering responsibilities closes the domain. Read it with DE-3 Reliability, incident response, and operational ownership, because a security incident follows the incident structure set out there.

Across domains

Technical judgment depends on Strategy & Vision for the organisation that an architecture implies, in SV-6 Organisational design and leading change, and for the economics of every platform and AI decision, in SV-4 Engineering economics. It depends on Delivery & Execution for the flow that review belongs to, DE-5 Designing how work flows, and for the risk and ownership gaps that surface as incidents, DE-4 Managing scope, risk, and dependencies across workstreams. The restraint that lets a team own its decisions is the same restraint PL-1 Delegation, ownership, and decision rights describes. When a board wants a headcount plan the evidence does not support, the honest answer is the one in CC-5 Values, integrity, and ethical judgment under pressure.

TJ-1 Evaluating architectural decisions and their organisational consequences

trade-offs, reversibility, and who has to coordinate because of this

1. Definition and why it matters

Managers rarely design systems, but they need to be in the room when systems are designed, because architecture has organisational consequences that outlast any release. Systems mirror the communication structures of the organisations that build them. Architecture and team design are therefore two views of one decision, and the most expensive architectural mistakes are organisational ones that nobody named. The competency is evaluating decisions rather than making them. That means weighing trade-offs and reversibility, spending deliberation on the one-way doors (data models, public contracts, core technology bets) while reversible choices move fast, and asking the questions a manager is uniquely placed to ask. What does this assume? What happens when it fails? Who has to coordinate because of this? It matters for two reasons. A manager who rubber-stamps architecture without engaging with it has handed the shape of the organisation to whoever designed the system. And at scale, the platform's runway is a business-continuity question that arrives disguised as an engineering one. The examinations test it through a rewrite proposed on charisma, two teams building the same thing twice, an eighteen-month warning nobody can see, and principal engineers whose deadlock has become an executive problem.

2. Core principles

1. **Evaluate for trade-offs and reversibility, not elegance.** Ask what the decision gives up and whether it can be undone. Scrutiny goes on the irreversible, and the reversible is allowed to move.
2. **Ask the questions only a manager will ask.** What does this assume? What happens when it fails? Who has to coordinate because of this? The whole posture is to be present in the design room without taking it over.
3. **A team boundary and a system boundary are one decision.** Design them together. Whichever one you leave undesigned will end up reshaping the other.
4. **Converge deliberately where convergence matters.** Across autonomous teams, standards live at the boundaries, and cross-cutting decisions are made once, on evidence, by a named process. They are not left to whoever argues longest or to parallel enthusiasm.
5.

Prefer the incremental path to the rewrite. Big-bang rewrites are how organisations re-learn old lessons at premium prices. Before a from-scratch proposal is funded, the incremental alternative has to be explored in earnest.

6. **Treat platform runway as business risk.** A warning that the core platform will not scale past the next eighteen months is a strategic decision, to be validated, sized and staged at the executive table. It does not become a backlog item because customers cannot see it.

---

3. Models and evidence

Architecture has one of the few empirically tested organisational claims in this standard, and the unit grades it as research.

Conway’s law [practice]

Conway observed in Melvin E. Conway, How Do Committees Invent? (1968) that a system's structure mirrors the communication structure of the organisation that built it. This unit uses the observation to evaluate designs. Any proposed architecture implies a coordination structure, so the manager asks who has to talk to whom because of the design, and whether the organisation is shaped to allow it. The practical corollary, designing the organisation to induce the architecture you want, is the design form covered in SV-6 Organisational design and leading change.

The mirroring hypothesis [research]

Alan MacCormack, Carliss Baldwin and John Rusnak, Exploring the Duality between Product and Organizational Architectures: The Mirroring Hypothesis (2012) tested Conway's observation empirically across paired software products. Products built by loosely coupled organisations were significantly more modular than functionally equivalent products built by tightly coupled ones. This is the evidence that the correlation Conway described is real and large, and it is why this standard treats the team boundary and the system boundary as one decision and not as a metaphor.

One-way and two-way doors [practice]

Jeff Bezos's classification in Jeff Bezos, 2015 Letter to Shareholders (Amazon.com, Inc.) (2016), used here to decide where architectural scrutiny goes. Data models, public contracts and core technology bets are one-way doors and deserve deliberation and senior involvement. Most other decisions are reversible and should be made quickly by the people closest to them. Reversibility is often misjudged, because "temporary" choices calcify, so a door is treated as one-way until shown otherwise.

The strangler pattern [practice]

Martin Fowler, StranglerFigApplication (2004) describes a way to replace a legacy system incrementally. New capability grows around the old system and routes traffic away from it piece by piece, until the old system can be retired. It is the standing alternative to the big-bang rewrite. Because the pattern exists, is well understood and lets reality vote early, a manager can demand that the incremental path be explored before a from-scratch proposal is funded.

Architecture review forums [practice]

A standing forum where significant designs get early scrutiny from other teams. It is most valuable when it advises, teaches and catches cross-team consequences, and least valuable as a gate every change must pass. Its provenance is ordinary practice. It fails by turning into a bottleneck and a rubber stamp as soon as it reviews everything, so its scope is limited to decisions with cross-team or irreversible consequences.

---

4. Practice

The three questions

In every significant design review the manager asks, in some form, what the design assumes, what happens when it fails, and who has to coordinate because of it. These are put as questions, not as objections. The answers improve the design; the manager's own design does not replace it.

The reversibility sort

Each design decision is labelled before it is made, as one-way or two-way. The team makes two-way decisions quickly and revisits them if they turn out wrong. One-way decisions get the deliberation, the review forum and the senior involvement. A team that gives every decision the same weight ends up slow on the trivial and careless on the irreversible.

The convergence decision

Sometimes two teams are about to build the same thing, or a cross-cutting choice such as a framework migration, a repository structure or a shared platform is being argued over. The manager then puts a decision in place, with criteria, the evidence required (a bounded pilot, a cost-benefit across every affected team), a named decider and a date. The organisation ends up with one solution or an explicit reason for two, and never with two by accident.

The consumer’s protection

If a team wants to rewrite a service that other teams depend on, the manager makes sure those consumers are protected: a stable contract through the transition, a migration plan agreed with the dependent teams, and compatibility guarantees. Whether to rewrite is the owning team's decision. Keeping its consumers stable is the manager's.

The runway review

Once or twice a year, at executive scope, the platform's runway is checked against the business's growth assumptions. What breaks first under the planned load? How long does the fix take? What would it cost to start now compared with later? The result goes to the executive table as a business-continuity decision with staged options.

---

5. Scaling note

At team scope the object is one team's design decisions and how reversible they are, and the manager improves them through questions and restraint. At organisational scope it becomes architecture across autonomous teams, meaning standards at the boundaries, deliberate platform bets and cross-cutting decisions made once on evidence. Here the manager makes sure the organisation converges where convergence matters without personally approving every design. At executive scope the object is technical strategy. That covers bet-the-company decisions, the platform's runway treated as business risk, platform sprawl consolidated on de-politicised criteria, acquired systems integrated by explicit decision, and deadlocks between experts turned into decisions. How Judgment Scales sets out the pattern; designing the organisation to induce the architecture is SV-6 Organisational design and leading change.

---

6. Judgment

- Is present for major design decisions and improves them through questions, without taking them over. [team]

TECHNICAL JUDGMENT

- Weighs organisational and long-term consequences as well as elegance. [team]
- Spends scrutiny on the irreversible and lets reversible decisions move. [team]
- Sees a team boundary and a system boundary as one decision.
- Failure mode: rubber-stamping architecture you do not engage with. [team]
- Failure mode: over-indexing on elegance while ignoring coordination cost. [team]
- Failure mode: treating all decisions as equally weighty. [team]
- Keeps architecture coherent across autonomous teams through standards at the boundaries and a review forum that advises and teaches instead of gating. [org]
- Decides a department-wide migration on evidence (a bounded pilot, a cost-benefit across every team, criteria) and not on advocacy or seniority. [org]
- Forces one deliberate decision when two teams are building the same thing. [org]
- Sets the criteria, evidence and decider for a cross-cutting choice such as a monorepo, and makes the department-level call. [org]
- Makes sure a dependent team is protected through another team's rewrite with a stable contract, an agreed migration and compatibility guarantees. [org]
- Builds a shared platform for proven demand, meaning duplication pain across several teams, and not for speculative demand, with ongoing ownership costed. [org]
- Failure mode: approving everything, which is a bottleneck, or nothing, which is drift. [org]
- Failure mode: letting parallel teams build the same thing twice. [org]
- Failure mode: a review board that gatekeeps instead of teaching. [org]
- Failure mode: platformising a guess. [org]
- Treats a principal engineer's eighteen-month platform warning as a strategic risk: validates it, sizes the bet, and takes it to the executive table with staged options. [exec]
- Answers a from-scratch rewrite proposal with deep scepticism, expressed as a demand for evidence and for the incremental alternative to be explored properly. [exec]
- Stages bet-the-company technical decisions so reality can vote early, sizes the downside for survivability, and gives the biggest bets the most scrutiny. [exec]
- Consolidates platform sprawl by de-politicising the decision, with objective criteria, honest migration accounting, clear ownership and respect for the people on the displaced platforms. [exec]
- Keeps technical coherence across acquisitions through explicit integration decisions instead of hoping for convergence. [exec]
- Converts a principal-engineer deadlock into a decision: frames the criteria, forces the honest comparison, and decides or explicitly delegates. [exec]
-

Failure mode: discounting the eighteen-month warning because customers cannot see it. [exec]

- Failure mode: funding a from-scratch rewrite on architectural charisma. [exec]
- Failure mode: letting five platforms live because each has a sponsor. [exec]
- Failure mode: waiting out a principal-engineer stalemate. [exec]

7. Tensions

Presence versus ownership. A manager in the design room improves the design. A manager who decides the design has taken it from the people who will live with it. Asking questions resolves this, because questions add considerations without removing ownership.

Coherence versus autonomy. Standards at every layer produce a uniformity that mostly serves dashboards. With no standards at all, you get five platforms and duplicated services. The line sits at the boundary: converge where teams meet and leave the inside to the team.

Deliberation versus speed. Scrutiny slows decisions down, and the irreversible ones deserve it. The reversibility sort lets the organisation move fast on most decisions and carefully on the few, instead of slowly on all of them.

Incremental versus clean. The strangler path takes longer to reach a clean end state, and it lets reality vote along the way. The rewrite promises a clean end state and usually arrives late, at a premium price, having re-learned the old system's lessons. The judgment is to insist the incremental path is explored honestly, and to fund the rewrite only when that exploration has failed in earnest.

Visible versus invisible risk. Customers see features. They do not see the platform's runway, and the business funds what it sees. At scope, the manager's job is to make the invisible risk visible in business terms before it turns into a visible outage.

8. Worked scenario

In a one-on-one, a principal engineer tells a head of engineering that the core data platform will not scale past roughly eighteen months of the company's planned growth. Fixing it means re-architecting the storage and event layers. The work would run for several quarters, involve four teams and be entirely invisible to customers. The product organisation has a full roadmap, and the board has been shown growth targets that assume the platform will keep working. The principal engineer has raised this twice before with their own manager, who filed it as technical debt.

There are two easy responses. One is to file it again, since nothing is broken and customers cannot see it. The other is to panic and commit the organisation to a re-architecture on the strength of one conversation. The first accepts an existential risk by default. The second bets four teams on a claim nobody has validated.

The head of engineering handles it as a strategic risk decision. The first step is validation. They ask the principal engineer and two others to produce the evidence within two weeks (the load model, what breaks first, at what growth, and how confident they are) and they read it themselves instead of accepting a summary. The claim holds, with a range of somewhere between fourteen and twenty months. The second step is sizing, and three options are costed. A full re-architecture now would take three quarters and displace two roadmap initiatives. A staged version would replace the event layer first, buying an estimated year of runway, and defer the storage work to a second stage with a checkpoint. A targeted set of mitigations would buy six months and no more. In the third step the decision goes where it belongs, to the executive table. It is framed as business continuity, since the growth plan the board has seen depends on a platform that will not carry it, and it arrives with the three options, their cost to the roadmap and a recommendation for the staged path.

The executive team chooses the staged path, and the two roadmap initiatives it displaces are named explicitly. The head of engineering then does the organisational half of the decision. Ownership of the event layer will move, so the four teams' boundaries are examined against the new architecture; a team boundary and a system boundary are one decision. The principal engineer is asked to lead the technical design, and an architecture forum reviews the one-way doors.

The head of engineering avoids both of the moves that would have felt like management: leaving the warning in the backlog because it was invisible, and deciding the architecture personally.

9. Related competencies

- SV-6 Organisational design and leading change: designing the organisation to induce the architecture, the inverse of this unit's evaluation.
- TJ-2 Technical debt: technical debt and systemic risk as the accumulated cost of architectural decisions.
- TJ-4 Technical credibility and intervention: knowing which architectural decisions are yours and staying close enough to judge them.
- DE-4 Managing scope, risk, and dependencies across workstreams: the coordination cost and boundary friction an architecture creates.
- SV-4 Engineering economics: the economics of a platform bet and the build, buy or partner decision behind it.

10. Self-check

1. Why focus architectural scrutiny on reversibility? *Answer:* Deliberation is expensive, and most decisions can be undone cheaply. Spending it on the one-way doors (data models, public contracts, core technology bets) and letting the two-way doors move keeps the organisation fast on most decisions and careful on the few that matter.
2. What are the three questions a manager is uniquely placed to ask in a design review? *Answer:* What does this assume? What happens when it fails? Who has to coordinate because of this? They bring in considerations the designers may not see, without taking the design away from them.
3. Two teams are building duplicate services for the same problem. What exactly is your role, and what would abdicating it look like? [org] *Answer:* To force one deliberate decision, with criteria, evidence, a decider and a date, so the organisation gets one solution or an explicit reason for two. Abdicating looks like letting parallel enthusiasm decide and then discovering two services in production.
4. What does the mirroring hypothesis add to Conway's law? *Answer:* Empirical evidence. Products built by loosely coupled organisations were measurably more modular than equivalent products built by tightly coupled ones, which turns an observation into a finding and a metaphor into a design rule.
5. A senior engineer proposes a department-wide framework migration costing three weeks per team; your managers are split. How is it decided? [org] *Answer:* On evidence: a bounded pilot, an explicit cost-benefit across all the affected teams, and a call made against criteria. Advocacy, seniority and who argues longest do not decide it.
6. A principal engineer says the platform dies in eighteen months. What makes your handling of this a strategy question rather than a backlog question? [exec] *Answer:* The growth plan the business has committed to depends on the platform carrying

it, so the risk is to business continuity. It is validated, sized into staged options with their roadmap costs, and decided at the executive table. It is neither buried because customers cannot see it nor panic-committed on the back of one conversation.

7. A respected architect proposes rewriting a large part of the core systems from scratch. What is the seasoned response? [exec] *Answer:* Deep scepticism, expressed as a demand for evidence and for the incremental alternative (the strangler path) to be explored properly first. Big-bang rewrites are how organisations re-learn old lessons at premium prices.

Sources

- Melvin E. Conway, How Do Committees Invent? (1968): systems mirror the organisations that build them.
- Alan MacCormack, Carliss Baldwin and John Rusnak, Exploring the Duality between Product and Organizational Architectures: The Mirroring Hypothesis (2012): the mirroring hypothesis tested empirically.
- Jeff Bezos, 2015 Letter to Shareholders (Amazon.com, Inc.) (2016): one-way and two-way doors.
- Martin Fowler, StranglerFigApplication (2004): the strangler pattern for incremental replacement.

TJ-2 Technical debt

recognising, quantifying, and scheduling repayment

1. Definition and why it matters

Technical debt is the implied cost of future rework that comes from choosing expedience now. A controlled amount is a sensible trade. Left unmanaged, the interest (slower delivery, more defects, more toil) compounds until movement stops. The competency is recognising debt for what it is, quantifying its cost in delivery and risk terms, and scheduling its repayment as ordinary work instead of as a crisis or a rewrite. As scope widens, the same disciplines govern debt's larger siblings: systemic risk that no single team can see, the part of the system everyone is quietly afraid to change, and knowledge that lives in one person's head. It matters because debt stays invisible until suddenly it is not, because the case for repaying it is usually made in the wrong terms, and because a manager who cannot price debt will be told the business cannot afford to pay it. The examinations test it through a product partner who wants no slowdown, a senior engineer who proposes the rewrite, a sole expert leaving in two months, and a board that wants to hear the incident will never recur.

2. Core principles

1. **Distinguish deliberate, prudent debt from reckless mess.** Debt taken on knowingly, for a reason and with the cost understood, is a trade. Debt nobody chose is mess with a flattering name, and it is managed differently.
2. **Make it visible and price it.** Debt is argued in delivery and risk terms ("this module causes a third of our incidents", "changes here take three times as long") and never on aesthetics. Debt that stays invisible gets repaid only after velocity collapses.
3. **Prioritise by interest rate.** Weigh the impact on delivery and risk against the cost of repayment. The highest-interest debt goes first, whatever happens to be most annoying.
4. **Repay incrementally, in the areas you already touch.** Continuous repayment alongside feature work beats both the mythical "later" and the big-bang rewrite, which is a high-risk way to re-learn old lessons.
5. **Read the systemic signals before any one team's metrics scream.** Incident rates trending up across teams, lead times stretching and knowledge concentrating in single heads all justify action at organisational scope before any team asks.
6.

Elevate risk to the level that should decide it. When debt threatens the business and product resists any slowdown, the manager stops negotiating capacity team by team. The risk goes to the executive table as a business-continuity decision with priced options. If feature pressure quietly accepts an existential risk, that is still a decision, and a badly made one.

---

3. Models and evidence

Technical debt began as a metaphor with a clear origin, and a practitioner vocabulary grew up after it. The unit grades it on that basis.

Debt as interest [practice]

The metaphor was coined in Ward Cunningham, The WyCash Portfolio Management System (1992). Shipping expedient code is like borrowing: it lets you move faster now, and interest is paid on every later change until the principal is repaid. For managers its value is that it turns the cost of expedience into a running cost instead of a one-off. Its limit is that it invites the wrong argument. Not all debt should be repaid, and the metaphor says nothing about which.

The technical-debt quadrant [practice]

Martin Fowler, TechnicalDebtQuadrant (2009) classifies debt as deliberate or inadvertent, and as prudent or reckless. Deliberate, prudent debt ("we know this is a shortcut, we will fix it after launch") is a sensible trade. Reckless debt is mess. Inadvertent, prudent debt ("now we know how we should have done it") is what learning ordinarily leaves behind. The vocabulary lets a team classify before it argues. Classifying changes nothing on its own; debt management is the visible tracking, priced impact and scheduled repayment built on top of it.

Continuous incremental repayment [practice]

The discipline of improving the structure of code in small, behaviour-preserving steps as part of ordinary work, in the areas being changed anyway, instead of in a separate rewrite. Martin Fowler, Refactoring: Improving the Design of Existing Code (Second Edition) (2018) gives it its technical form. Its importance to managers is economic. Repayment in areas already being touched costs almost nothing, and a standing share of capacity for the highest-interest debt elsewhere is cheap. The rewrite is the most expensive and riskiest form of repayment there is.

Systemic risk signals [practice]

This is an organisational-scope practice that the unit treats as central, though it is not a named model. The manager reads signals that no single team can see and acts on them before a team's own metrics force the issue. The signals include cross-team incident trends, stretching lead times, change-failure rates in particular areas, the module everyone avoids, and the one expert who understands a given system. The practice fails when the manager waits for a team to ask.

Elevation to business risk [practice]

Also not a named model. At executive scope, technical risk is converted into business terms and decided at the level it deserves: platform debt as continuity risk, a vulnerability class as exposure, and priced options with an honest roadmap impact. Its evidence is experiential. Its failure is the most common one at scope, where capacity is negotiated team by team while the organisation accepts an existential risk by default.

---

4. Practice

The debt register with prices

The team keeps a short list of its significant debt. Each item carries its impact in delivery or risk terms (incidents caused, time added to changes, features blocked) and an estimate of what repayment would cost. The list is the argument. It turns "the code is bad" into "this module causes a third of our incidents".

The touch rule

Any change to an area with known debt repays some of it. The rule is small and continuous, and it keeps the register from growing. It costs almost nothing because the engineer is already working there.

The highest-interest allocation

The standing capacity share from SV-3 Roadmapping goes on the register's highest-interest items, ranked by impact against cost. The most annoying item, the oldest one and the one an engineer most wants to rewrite do not get it.

The rewrite challenge

TECHNICAL JUDGMENT

A proposal to rewrite a working system is treated as a bet and evaluated as one. What would the incremental alternative cost? What evidence says the current system cannot be improved in place? What happens if the rewrite takes twice as long? The incremental path is explored honestly before the rewrite is funded. Most rewrites do not survive the challenge, and the ones that do are worth funding.

The expiring-option review

When the team discovers a departing expert, a fragile vendor or a single point of failure, the response starts that week, with pairing, documentation, shadowing or redundancy. The chance to act cheaply runs out on a known date, and that date is why it outranks most roadmap items.

5. Scaling note

At team scope the object is the team's own debt and its repayment schedule. The manager keeps it visible, prices it and repays it incrementally. At organisational scope the object becomes systemic debt that no single team owns, along with the risk signals no single team can see. The manager reads those signals early, funds class-level fixes with named owners, evaluates a large refactor on proxy evidence and stages it to prove value, and acts on expiring options as soon as they are found. At executive scope debt is a strategic liability (the platform's runway, the business's exposure) and is decided at the executive table in business terms. Commitments to the board take the form of measurable remediation programmes instead of promises, and resilience is verified instead of assumed. How Judgment Scales sets out the pattern; the architectural decisions that create debt are TJ-1 Evaluating architectural decisions and their organisational consequences.

6. Judgment

- Distinguishes prudent debt from reckless mess before arguing about either. [team]
- Keeps debt visible, with its impact priced in delivery and risk terms. [team]
- Repays continuously, in the areas already being touched, alongside feature work. [team]
- Treats a proposed rewrite as a bet and demands the incremental alternative be explored first.
- Failure mode: invisible accumulation until velocity collapses. [team]
- Failure mode: aesthetic arguments for repayment. [team]
- Failure mode: the mythical "later". [team]

TECHNICAL JUDGMENT

- Failure mode: the big-bang rewrite. [team]
- Reads cross-team risk signals (incident trends, stretching lead times, feared areas, concentrated knowledge) before any team's metrics scream. [org]
- When debt is slowing delivery and product wants no slowdown, states the honest trade-off with evidence: the pace is borrowed from future quarters, this is the measured cost, and this is the proposal. [org]
- Evaluates a large refactor with a hard-to-quantify payoff on proxy evidence, then stages and time-boxes it so it proves its value as it goes. [org]
- Assigns a named owner and a class-level fix when the same gap appears across several teams. [org]
- Makes knowledge transfer the immediate priority when a sole expert is leaving, because the option expires in weeks. [org]
- Treats a vendor outage that took down several teams as the discovery of a systemic dependency, and weighs the cost of resilience against the measured blast radius. [org]
- Funds the department's immune system: testing and continuous integration, observability, blameless reviews, dependency hygiene and continuous refactoring. [org]
- Failure mode: waiting for a team to ask before seeing the systemic trend. [org]
- Failure mode: letting "no feature slowdown" stand as a decision nobody made. [org]
- Failure mode: patching a cross-team finding one instance at a time. [org]
- Failure mode: discovering the bus factor at the leaving party. [org]
- Takes debt that threatens the business to the executive table as a continuity decision with priced options, and stops negotiating capacity team by team. [exec]
- Prioritises a vulnerability class that spans many services as business risk, with explicit capacity and an honest roadmap impact, instead of squeezing it into team slack. [exec]
- After a major incident, commits to the board a systemic remediation programme with measurable milestones and honest timelines, and never "it won't happen again". [exec]
- Can show with evidence whether the organisation survives its plausible bad days: single points of failure mapped, degradation graceful, recovery tested, key dependencies bounded. [exec]
- Personally tracks a short list of risks: platform runway, security posture, reliability trends and concentration risks. [exec]
- Failure mode: letting product pressure quietly accept existential risk. [exec]
- Failure mode: promising the board recurrence-free operation. [exec]
- Failure mode: treating security capacity as something teams find down the back of the sofa. [exec]
-

TECHNICAL JUDGMENT

Failure mode: delegating reliability so completely that the trend arrives as the outage. [exec]

7. Tensions

Expedience versus interest. Taking on debt knowingly is often right, when the launch matters and the shortcut is understood. The trouble is that a future team, often a different one, pays the interest. The judgment lies in writing the loan down so that it gets repaid instead of discovered.

Visibility versus alarm. A long debt register makes a team look as if it is in trouble, while a short one hides the interest. Keep the register honest and priced, and frame it as control: this is what it costs, and this is the schedule.

Incremental versus rewrite. The strangler path is slower to reach a clean state. The rewrite reaches a promise faster and delivery more slowly. The judgment is to demand that the incremental path be explored properly, and to fund the rewrite only when that exploration fails on evidence.

Repayment versus delivery. Every hour spent on debt is an hour not spent on features this quarter, and every hour not spent on debt is borrowed from a future quarter at interest. A stated capacity share resolves this in principle. The hard part is holding to it when a launch is close.

Honesty versus reassurance at the board. After a serious incident the board wants to hear that it cannot happen again, and the honest commitment is a measurable programme with timelines. Reassurance is discredited the moment the next incident happens. The programme is believable because the board can check it.

8. Worked scenario

During sprint planning, a senior engineer proposes to the team's manager that the order-processing service be rewritten from scratch. The service is six years old, written in a framework the team no longer likes, and the source of roughly a third of the team's incidents. The engineer estimates the rewrite at one quarter with three people. The rest of the team is keen. The product partner is not in the room, and the current roadmap assumes the whole team is on features.

The two easy answers are to approve it, because the team is right that the service is a problem and the engineer is capable, or to refuse it, because the roadmap is full and rewrites are dangerous. Neither deals with the real question, which is what the debt costs and what the cheapest way to repay it would be.

The manager starts with the register. The incident data supports the engineer's claim about a third of incidents. It also shows that nearly all of them come from two modules, the inventory reconciliation and the retry logic, and not from the framework. Changes to those two modules take three times as long as changes elsewhere. The rest of the service is old, unloved and rarely touched, with an unremarkable incident rate. The debt is real, priced and concentrated.

The manager then puts the rewrite through the challenge. The incremental alternative is to replace the two modules in place, behind their existing interfaces, with one engineer over six weeks, and leave the rest of the service alone. That addresses the priced cost at a fraction of the price, and reality gets a vote after the first module. The rewrite would replace four years of accumulated edge-case handling in the parts of the service that work, on an estimate that, as DE-2 Estimation, planning, and honest commitments would predict, is the optimistic one. The manager says this plainly, with the numbers, and proposes the incremental path, funded from the team's standing debt share and starting with the reconciliation module.

The senior engineer is disappointed, having wanted the clean system. The manager gives them the lead on the two modules and says what would change the answer. If, once both modules are replaced, the incident and change-time data still point at the service, the rewrite conversation reopens with that evidence. The product partner is told, in exactly these terms, that the team's debt share for the next six weeks is going to the thing that causes a third of the incidents.

The manager does not fund a quarter-long bet on the strength of a capable engineer's enthusiasm. Nor do they dismiss a real problem because the proposed solution was the wrong one.

9. Related competencies

- SV-3 Roadmapping: the standing capacity share that funds repayment, and the roadmap it competes with.
- TJ-1 Evaluating architectural decisions and their organisational consequences: the architectural decisions that create debt, and the rewrite set against the strangler path.
- DE-4 Managing scope, risk, and dependencies across workstreams: key-person risk and vendor dependency as systemic risk.
- TJ-6 Security, privacy, and compliance as engineering responsibilities: security exposure as debt's sibling, prioritised the same way.
- SV-2 Communicating and partnering across functions: making the case on debt to product and to the board in their terms.

10. Self-check

1. What makes debt "prudent", and what repayment pattern beats a rewrite? *Answer:* Prudent debt is taken on deliberately, for a reason, with the cost understood. Continuous incremental repayment in the areas already being touched, plus a standing share for the highest-interest items elsewhere, beats the big-bang rewrite, which is the most expensive and riskiest form of repayment.
2. How is debt argued to a product partner? *Answer:* In delivery and risk terms, such as incidents caused, time added to changes and features blocked, and never on aesthetics. The priced register is the argument.
3. A senior engineer proposes rewriting a working system. What is it treated as, and what must happen first? *Answer:* A bet, and it is evaluated as one. The incremental alternative is explored honestly first (where the debt is concentrated, and what replacing that part in place would cost), and the rewrite is funded only if that exploration fails on evidence.
4. A departing expert holds a legacy system alone. Why does this outrank most roadmap items right now? [org] *Answer:* The option to act cheaply, through pairing, documentation and shadowing, expires on a known date. After that, the same knowledge costs an order of magnitude more to recover, if it can be recovered at all.
5. Debt is slowing delivery across the organisation and product wants no drop in feature output. What is your position? [org] *Answer:* The honest trade-off, with evidence. The current pace is being borrowed from future quarters; this is the measured cost; this is a proposal for an explicit capacity share aimed at the highest-interest debt. The decision is made consciously, above you if necessary, and is not left to be settled by silence.
6. The board wants assurance that a major incident will not recur. What can you honestly commit to, and why is "never again" disqualifying? [exec] *Answer:* A systemic remediation programme with measurable milestones and honest timelines. "Never again" is false, since no system is free of incidents, and the next incident discredits everything else you say.
7. What does an executive keep on a personally tracked risk list, and why not delegate it entirely? [exec] *Answer:* Platform runway, security posture, reliability trends and concentration risks. Reliability and technical risk are strategic matters. An executive may operationalise them through others but should never fully delegate them, because the alternative is learning about the trend from the outage.

Sources

TECHNICAL JUDGMENT

- Ward Cunningham, The WyCash Portfolio Management System (1992): the debt metaphor and its interest.
- Martin Fowler, TechnicalDebtQuadrant (2009): deliberate or inadvertent, prudent or reckless.
- Martin Fowler, Refactoring: Improving the Design of Existing Code (Second Edition) (2018): incremental, behaviour-preserving improvement as ordinary work.

TJ-3 Code review and quality standards

including standards for AI-generated code

1. Definition and why it matters

Code review is one of the most powerful quality and culture mechanisms a team has. Done well, it catches defects, spreads knowledge and raises the shared standard. Done badly, it becomes a bottleneck, a battleground or a rubber stamp. Quality is broader than review. It is the whole system of tests, continuous integration, guardrails and standards that makes the right thing the easy thing, and at every scope the manager's job is to invest in that system instead of inspecting every change. v3.0 adds a question v1.0 could not have asked: what must review establish when the author did not write every line? AI assistance has increased the volume of code and made the author's understanding of it less certain, and a review practice built for human-written changes will pass generated code it should stop. This matters because the quality bar is what a team does not trade under deadline pressure (scope is what gets traded), and because the most common form of quality failure is a standard that exists on paper while pull requests are waved through. The examinations test it through a deadline that tempts the team to skip tests, review that has grown cursory as assistants made changes larger, a breaking API left as the consumers' problem, and a niche stack nobody else can maintain.

2. Core principles

1. **Trade scope, never the tests.** Under deadline pressure the quality bar holds and the scope moves. The tests are how the team knows the smaller thing works.
2. **Review is timely, kind, specific and proportionate.** Slow review blocks whole teams, and unkind review teaches people to hide their work. Attention belongs on design, clarity and intent, and trivia should never distract a reviewer from risky logic.
3. **Automate the objective, so people judge the subjective.** Formatting, linting and tests belong to the pipeline. Human review is for what only humans can judge.
4. **Coverage is not quality.** Tests that catch what matters are quality. More tests of the wrong kind add drag without adding confidence.
5. **Review establishes understanding, not just correctness, when code is generated.** Whoever submits a change owns it and can explain it. Generated code is verified, tested and disclosed, and its security and licensing are checked as a class instead of being assumed.
6.

> **Standards are a floor at the boundaries, not a uniform ceiling.** The floor covers interface contracts, quality bars, security baselines and operational readiness, which is where teams meet. Inside a team, tools and style belong to the team, weighed on their organisational cost.

---

3. Models and evidence

Code review is among the better-studied practices in software engineering, and two of the unit's claims rest on research-grade evidence.

Code review as knowledge transfer [research]

Alberto Bacchelli and Christian Bird, Expectations, Outcomes, and Challenges of Modern Code Review (2013) studied review at scale in one company. Developers and managers expected review to find defects, but its observed benefits were as much knowledge transfer, team awareness and better solutions as defect detection, and the main challenge reviewers faced was understanding the change. Two consequences follow for managers. Review is a teaching mechanism and should be designed as one. And a review of a change the reviewer does not understand has delivered none of its benefits, which is why generated code needs review that establishes understanding.

Security of AI-generated code [research]

Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt and Ramesh Karri, Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions (2022) found that a substantial share of the code an AI assistant produced, across a set of security-relevant scenarios, contained vulnerabilities of known classes. Later tools and models change the proportion. The structural point stands: generated code has its own vulnerability profile, which has to be checked as a class through scanning and review aimed at the known failure modes. Fluency is no evidence that it is safe. The governance consequences are covered in TJ-5 AI-assisted engineering and TJ-6 Security, privacy, and compliance as engineering responsibilities.

Review for AI-generated code [practice]

This is not a named model. It is the standard this unit proposes for what review must establish when an assistant produced some or all of a change. The submitter understands the change and can explain it. It is tested at the level the risk warrants. Its origin is disclosed where that matters. It has been checked against the vulnerability classes generated code is prone to. And it does not bring in code under a licence the organisation cannot accept.

The standard is set explicitly, before adoption, and it is proportionate, so a documentation change is not checked the way a payment path is.

Quality as a system [practice]

Also not a named model. It is the stance this unit treats as the manager's real job: tests, continuous integration, guardrails, standards and the paved road form a system that makes the right thing the easy thing, and the manager invests in it instead of inspecting it. Its provenance is ordinary practice. It fails when a manager reviews every change personally and calls that rigour.

Contract at the boundary [practice]

The practice comes from DE-4 Managing scope, risk, and dependencies across workstreams, and Matthew Skelton and Manuel Pais, Team Topologies: Organizing Business and Technology Teams for Fast Flow (2019) gives it its team-level form. Interface discipline (versioning, a deprecation policy, contract tests, communicated change) is the quality floor wherever work crosses a team boundary. Consumers' stability is part of the owning team's definition of done. A widely used interface that keeps breaking its consumers is a quality failure of the team that owns it, and the consumers should not be left to absorb it.

---

4. Practice

The review service level

The team agrees how quickly a review gets answered (the same day, for most changes) and measures it. Review latency is a flow metric. A pull request that waits three days blocks its author and usually someone else as well.

The proportionality rule

Review attention goes where the risk is: design, intent, and the paths that touch money, data or security. Formatting and style are automated. A reviewer who leaves fifteen comments on naming and none on the retry logic has reviewed the wrong thing.

The generated-code checklist

Before AI tooling is adopted, the team writes down what review must now establish. The submitter explains the change in their own words. Tests exist at the level the risk warrants. Provenance is disclosed where the organisation requires it. The change passes

the scanners configured for the vulnerability classes generated code is prone to, and no unacceptable licence has come in with it. The checklist is proportionate to the risk of the change, and wherever it can be, it is enforced in the pipeline.

Scope before tests

When a deadline is close and someone proposes skipping tests to make it, the manager reframes the question. What can be cut from the scope so that a smaller thing ships tested? The tests stay where they are.

The standards floor

A group of teams gets a short written floor at the boundaries, covering interface contracts, the definition of done where work crosses a team boundary, security baselines and operational readiness. Alongside it goes an explicit statement that tools, ceremonies and style inside a team are the team's own business. A team that wants a niche language is asked the organisational questions: maintainability, hiring, bus factor, and who supports it once the enthusiasts leave.

Service ownership

Every service has a named owner, responsible for its operation, interface stability, documentation and future. The ownership map is checked after every reorganisation. A critical service with no owner is the priority, because everything about its health depends on someone being responsible for it.

5. Scaling note

At team scope the object is the team's review practice and quality bar. The manager keeps reviews fast, kind and proportionate, automates the objective checks, sets the generated-code standard, and trades scope before tests. At organisational scope the object becomes consistent standards across autonomous teams, set without prescribing each team's tools: a floor at the boundaries, interface discipline for widely used APIs, no gaps in service ownership, and technology choices weighed on organisational cost. At executive scope the object is the organisation's quality posture and the systems that maintain it. That means a paved road that earns voluntary adoption, an explicit and bounded innovation budget with evaluation criteria, and the recognition that a pattern of quality failures across the organisation is a leadership failure before it is an engineering one. How Judgment Scales sets out the pattern; the governance of AI assistance is TJ-5 AI-assisted engineering.

6. Judgment

TECHNICAL JUDGMENT

- Keeps reviews fast, kind, specific and proportionate, and treats review as a way of sharing knowledge. [team]
- Automates objective checks so that human attention goes to design, clarity and intent. [team]
- Makes standards explicit and shared, so that "good" does not depend on which reviewer you draw. [team]
- Cuts scope instead of tests when a deadline tempts the team. [team]
- States what review must establish for AI-generated code, and who is accountable for it.
- Notices when assistants have made changes larger and faster than review can handle, and changes the review while keeping the bar.
- Failure mode: reviews that sit for days. [team]
- Failure mode: the rubber stamp. [team]
- Failure mode: bikeshedding trivia while risky logic goes through. [team]
- Failure mode: "temporarily" skipping tests under pressure. [team]
- Failure mode: passing generated code the submitter cannot explain.
- Sets a quality floor at the boundaries (interface contracts, a definition of done at the seams, security baselines, operational readiness) with autonomy inside. [org]
- Weighs a team's wish to adopt a niche stack on organisational cost: maintainability, hiring, bus factor, and support once the enthusiasts leave. [org]
- Holds the owner of a widely used API to interface discipline: versioning, a deprecation policy, contract tests and communicated change. [org]
- Makes sure no critical service lacks a named owner, and treats an orphaned one as the priority. [org]
- Sets standards for AI-generated code across teams, covering verification, ownership and disclosure, without prescribing each team's tools. [org]
- Failure mode: mandating uniformity where autonomy was the value. [org]
- Failure mode: waving through a niche stack the organisation cannot sustain. [org]
- Failure mode: letting a breaking API stay "the consumers' problem". [org]
- Failure mode: reorganisations that silently orphan services. [org]
- Failure mode: a definition of done nobody enforces at the seams.
- Steers quality and technology choice through a paved road that earns voluntary adoption, with teams free to leave it at their own cost. [exec]
- Holds an explicit, bounded innovation budget, with pilots that have criteria, a contained blast radius and an honest review. [exec]
-

TECHNICAL JUDGMENT

Migrates away from a fashionable adoption that became a liability with a deliberate plan and without blame, and sets adoption criteria before enthusiasm gets the next vote. [exec]

- Recognises a pattern of quality failures across the organisation as a leadership failure before it is an engineering one. [exec]
- Failure mode: freezing the organisation's stack in the name of consistency. [exec]
- Failure mode: letting every team be its own chief technology officer. [exec]
- Failure mode: adopting on hype and abandoning with the hangover. [exec]
- Failure mode: punishing the team that took a risk the organisation never bounded. [exec]

7. Tensions

Speed versus scrutiny. Fast review keeps the team flowing, and thorough review catches what matters. Proportionality settles it. The trivial change takes the fast path, the risky path gets the attention, and automation takes the objective checks off both.

Throughput versus understanding. AI assistance raises the volume of code and lowers the author's certainty about it. Review that keeps pace with the volume stops establishing understanding. The judgment is to hold review to understanding and let throughput fall where it falls, because code nobody understands is debt at its highest rate of interest.

Consistency versus fit. Standardising tools across teams is convenient for dashboards, and each team pays for it in fit. Leaving every choice to the teams produces stacks nobody else can maintain. The line is a floor at the boundaries, with organisational cost as the test for everything else.

Autonomy versus liability. A team that adopts a bleeding-edge technology has taken a risk the organisation may need taken. If it does not pay off, the same team is left with a maintenance and hiring liability. Bounding the risk in advance, with criteria, a blast radius and a review, lets the organisation take it without punishing the people who did.

Kindness versus candour in review. Kind review can drift into saying nothing, and candid review can turn into a battleground. The rule is to be specific and to comment on the code, never on the author. It is the same rule as in PL-2 Feedback and difficult conversations.

8. Worked scenario

Three months after the team adopted an AI coding assistant, its manager notices three things. Pull requests have become larger and more frequent, review turnaround has

dropped below an hour, and the number of comments per review has fallen by two-thirds. Delivery looks faster. Then a change to the payment retry logic ships with a subtle bug, a retry loop that could double-charge customers during a network partition. In the incident review the engineer who submitted it is asked how the loop works. They say the assistant generated it and it passed the tests.

The two easy responses are to ban the assistant, which throws away a real gain and will be quietly ignored, or to treat the incident as a one-off and remind everyone to be careful. Neither deals with what has changed. The review practice was built for changes their authors understood, and it is now passing changes their authors did not.

The manager changes the review and keeps the bar. With the team, they write down what review must now establish. The submitter explains the change in their own words in the pull request. Changes that touch money, data or security carry tests at the level of that risk, and the reviewer checks the tests as carefully as the code. The pipeline runs scanners configured for the vulnerability classes generated code is prone to. A reviewer who does not understand a change asks about it and withholds approval until they do. The manager also looks at the review service level and finds the real signal there: turnaround fell because reviewers stopped reading. They restore proportionality, so that payment paths get a named second reviewer and documentation changes get the fast path. They also measure understanding as well as speed, by sampling a few reviews a week and asking the reviewer to explain the change.

Nobody blames the engineer who shipped the bug in the incident review, because the system passed it. In a one-on-one they are asked to take proper ownership of the retry logic, and they do. The product partner is told that delivery will look slower for a few weeks, and why.

The manager does not accept throughput as the measure of the adoption. Fast delivery of code nobody understands is debt at its highest rate of interest, and the incident was the first payment.

9. Related competencies

- TJ-5 AI-assisted engineering: the adoption, governance and capability-building that decide what review has to cope with.
- DE-5 Designing how work flows: review as part of how work flows, and aligned autonomy as the principle behind standards.
- DE-4 Managing scope, risk, and dependencies across workstreams: contract at the boundary, and interface discipline between teams.
- TJ-6 Security, privacy, and compliance as engineering responsibilities: the security baseline in the standards floor, and vulnerability classes checked as a class.

- CC-1 Psychological safety and productive conflict: the safety that makes review a place to learn instead of a place to be judged.

10. Self-check

1. Why is scope the thing you trade under pressure, never the tests? *Answer:* The tests are how the team knows the smaller thing works. Skipping them to hit a date ships something nobody can vouch for. Cutting scope ships something smaller that is known to work.
2. What does the research say review actually delivers, beyond finding defects? *Answer:* Knowledge transfer, team awareness and better solutions. The main challenge reviewers face is understanding the change, and a review of a change the reviewer does not understand delivers none of these benefits.
3. What must review establish for AI-generated code? *Answer:* That the submitter understands the change and can explain it, that it is tested at the level the risk warrants, that its origin is disclosed where required, that it has been checked against the vulnerability classes generated code is prone to, and that no unacceptable licence came in with it. All of this is proportionate to the risk of the change.
4. What belongs in a cross-team standard, and why does "standardise everything" fail? [org] *Answer:* A floor at the boundaries: interface contracts, quality bars where work crosses teams, security baselines and operational readiness. Standardising every tool and language gives up local fit and ownership in exchange for a uniformity that mostly benefits dashboards.
5. A team wants to adopt a niche language only they know. What is the key consideration? [org] *Answer:* The organisational cost (maintainability, hiring, bus factor, and who supports it once the enthusiasts leave), weighed against the real technical benefit.
6. What does a paved road change about how quality and technology choice scale? [exec] *Answer:* Consistency comes from making the good path the easy one instead of from a mandate. Teams adopt the supported default voluntarily and may leave it if they take on the ownership cost themselves. A paved road nobody would choose is a mandate in costume.
7. A team's bleeding-edge adoption has become a maintenance and hiring liability. What is the action, and what is the lesson? [exec] *Answer:* A deliberate migration plan, without blame, because the organisation never bounded the risk. The lesson is that adoption criteria (fit, support, exit cost) have to exist before enthusiasm gets a vote.

Sources

TECHNICAL JUDGMENT

- Alberto Bacchelli and Christian Bird, Expectations, Outcomes, and Challenges of Modern Code Review (2013): what modern code review delivers in practice, and why understanding is its central challenge.
- Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt and Ramesh Karri, Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions (2022): the vulnerability profile of AI-generated code.
- Matthew Skelton and Manuel Pais, Team Topologies: Organizing Business and Technology Teams for Fast Flow (2019): interface discipline and contracts between teams.

TJ-4 Technical credibility and intervention

staying close enough to judge, and knowing when to step in

1. Definition and why it matters

v1.0 had two competencies here, staying technically credible and judging when to intervene. v3.0 merges them because they are one judgment. You can only decide when to step into a technical decision if you are close enough to see it clearly, and staying close is only worth what it costs because of the decisions it lets you make well. Once a manager stops coding daily, breadth fades slowly and depth fades fast. Systems understanding, which is the knowledge that matters for leading, persists if it is deliberately maintained. Credibility does not mean being the strongest coder. It means understanding the system well enough to ask sharp questions, to tell real risk from hypothetical risk, and to know whom to trust on what. Intervention is the hardest technical judgment because its default is restraint. At every scope a manager will see decisions heading somewhere they would not have taken them. Stepping in too readily robs people of ownership and learning, and failing to step in on a dangerous call is abdication. It matters because both failures are silent. The manager who has drifted away from the system does not know it, and the manager who overrides on preference ends up with a team that has stopped deciding. The examinations test it through a new manager who is still doing the coding, a team about to make a call its manager believes is wrong, an executive who has become the chief architect, and a strategic decision left to whoever felt strongly.

2. Core principles

1. **Maintain systems understanding, not currency.** Depth in the code fades, and you cannot keep it without doing the job you left. Your understanding of the system's shape, risks and dependencies will last if you read designs, stay in the hard rooms and go deliberately deep in one area at a time.
2. **Credibility is the quality of your questions, not the competition with the answers.** At every scope, honest deference ("you know this better than I do; walk me through the trade-off") builds more trust than pretending.
3. **Intervene when the cost of being wrong is high and hard to reverse.** That means one-way doors, safety, security, data integrity and customer impact in flight. Stay out when the decision is reversible and owning the outcome will teach the team more than your answer would.
4.

Intervene by raising what was missed, not by overriding with authority. The intervention adds considerations. The decision stays where it was.

5. **Above team scope, install a decision process before issuing an opinion.** Set the criteria, the evidence, a named decider and a date. A manager of managers who personally approves every architectural decision is a bottleneck, however diligent it looks, and teaches their managers not to decide.
6. **Keep an explicit, short docket of the decisions that are yours.** At executive scope a handful of technical decisions are inseparable from company strategy and are never fully delegated. Everything else is delegated with real authority, and restraint is a matter of policy instead of mood.

---

3. Models and evidence

There is little empirical literature on this competency and a great deal of practitioner consensus. The unit grades its models as practice, and borrows two of them from elsewhere in the standard.

Systems understanding over currency [practice]

This is the distinction the unit rests on, though it is not a named model. The knowledge that matters for leadership is the map: how the system is shaped, where its risks lie, what depends on what, and who understands each part. A manager can maintain it by reading designs and code without becoming an approver, by staying in the room for hard decisions, and by rotating deep dives. Depth in the code is a different thing, and trying to keep it while managing usually means doing both jobs badly. The argument is experiential. The failure it describes, expertise frozen at the manager's last hands-on year, is common enough to recognise.

Task-relevant maturity [practice]

Grove's concept from Andrew S. Grove, High Output Management (1983), applied here in reverse. The manager's own task-relevant maturity with the technology in front of them decays, and an honest manager calibrates their confidence to match. A manager whose understanding of the current system is in fact familiarity with the old one has low maturity and high confidence. That combination produces the nostalgia-driven intervention in a stack that has moved on.

The intervention threshold [practice]

This is the rule the unit adopts and the one the examinations test most directly, though again it is not a named model. Step in when the cost of being wrong is high and hard to reverse.

Stay out when the decision is reversible and owning it will teach more than the correction is worth. The rule has no author. It is Jeff Bezos, 2015 Letter to Shareholders (Amazon.com, Inc.) (2016)'s one-way and two-way doors put to practical use on the question of whose decision it is. Its two failures mirror each other: overriding on preference, and avoiding a dangerous call to dodge conflict. A team that suffers the first cannot make any decision, and a team that suffers the second makes a fatal one.

Installing a decision process [practice]

Intervention takes this form at organisational scope. A cross-team decision may be deadlocked, may be irreversible at department scale, or may quietly commit the whole department. The manager of managers then usually intervenes by installing the process that will produce the answer (the criteria, the evidence required, who decides and by when) instead of supplying the answer. The intervention becomes a capability the organisation keeps.

The executive docket [practice]

Not a named model either, but the executive-scope practice this unit treats as decision architecture. The executive keeps an explicit, short list of the technical decisions they own personally because those decisions are inseparable from company strategy. These are bet-the-company technology choices, build-versus-buy for the capability that most differentiates the product, vendor lock-in on a core capability, infrastructure the product will increasingly stand on, and the organisation's security and reliability posture. Everything else is delegated to senior engineers with real authority. A docket that is too long is micromanagement with a title. Without a docket, the strategic decisions get made by default.

---

4. Practice

Reading without gatekeeping

The manager reads design documents, and some code, to keep the map current, and is not a required approver of either. The reading turns into questions in the design room. It never turns into approvals in the pipeline.

The quarterly deep dive

Each quarter the manager picks one system or domain and goes deliberately deep into it: the design, the risks, the people who understand it, and what breaks first under load. Being

shallow everywhere is not understanding. Going deep somewhere, and rotating, keeps the map honest in the places that matter.

The threshold, stated

The manager writes down for the team the kinds of decision they will step into, such as the one-way doors and anything touching safety, security, data integrity or customers in flight, and says that everything else belongs to the team. The list is short and applied consistently, so the team can predict it.

Intervening as questions

When the manager does step in, they bring what was missed. "What happens to in-flight orders during the cutover?" is an intervention; "do it this way" is an override. If the question does not change the decision, the team owns the outcome and the manager owns the restraint.

The decision process, installed

When a cross-team decision is stuck or dangerous, the manager of managers sets out the process before offering any opinion: the criteria for judging the options, the evidence required, the person who will decide, and the date. Opinions come after the evidence.

The docket, written

The executive keeps a short written list of the technical decisions they own and reviews it twice a year. Everything not on it is delegated with real authority. When a docket decision is about to be made somewhere else (a differentiating capability about to be bought because one team preferred to buy it, say), the executive takes it back explicitly.

---

5. Scaling note

At team scope the question is when to step into a design or a fix and when not to. The manager stays close through reading and presence and owns the restraint. At organisational scope the manager stays credible at one level up. They review designs instead of code, rotate deep dives, use architecture forums and principal engineers as standing tutorials, and maintain a map of systems, risks and experts. They intervene by installing a decision process when scope, irreversibility, deadlock or a department-level commitment calls for it, and they act through their managers instead of around them. At executive scope credibility becomes structured sensing: architecture and operations reviews with evidence, principal engineers who tell the truth, and sampled raw signal that keeps the aggregates honest. Intervention becomes decision architecture, meaning the short docket and real delegation

of everything else. How Judgment Scales sets out the pattern; the economics of the docket's build-versus-buy decisions are in SV-4 Engineering economics.

6. Judgment

- Reads code and designs to stay close to how the system is evolving, without becoming a required approver. [team]
- Stays in the room for hard technical decisions. [team]
- Goes deliberately deep in one area each quarter instead of staying shallow everywhere. [team]
- Defers honestly to engineers' depth: "you know this better than I do; walk me through the trade-off."
- Lets reversible decisions ride even when they would choose differently, and steps in on the irreversible and the dangerous. [team]
- Intervenes by raising what was missed instead of overriding with authority.
- Failure mode: clinging to best-coder status and becoming the bottleneck. [team]
- Failure mode: faking understanding. [team]
- Failure mode: drifting so far from the system that you can no longer reason about it. [team]
- Failure mode: overriding on preference. [team]
- Failure mode: avoiding a dangerous call to dodge conflict. [team]
- Failure mode: being the permanent final approver. [team]
- Stays credible at one remove, with design reviews standing in for code reading, rotating deep dives, architecture forums and principal engineers used as tutorials, and a maintained map of systems, risks and experts. [org]
- Asks the questions that expose reasoning, such as what breaks first under ten times the load, and earns a principal engineer's respect for the questions instead of competing with the answers. [org]
- Gets directly involved when a decision crosses team boundaries with material consequences, is irreversible at department scale, is deadlocked at a cost greater than that of deciding, or quietly commits the whole department. [org]
- Installs a decision process (criteria, evidence, decider, date) before issuing an opinion. [org]
- Intervenes through the manager, and never around them, when a team is about to make a decision the manager of managers believes is wrong. [org]
- Failure mode: technical currency by nostalgia, frozen at the last hands-on year. [org]

- Failure mode: skipping the technical rooms entirely. [org]
- Failure mode: the approval bottleneck. [org]
- Failure mode: picking sides in a deadlock instead of forcing a fair decision. [org]
- Failure mode: finding out about a department-level commitment after the fact because "it was the team's call". [org]
- Maintains judgment through structured sensing: architecture and operations reviews with evidence, principal engineers who tell the truth, periodic deep dives into the systems that carry the most strategic risk, and external calibration. [exec]
- Keeps aggregate signals honest with sampled raw material, such as an incident review attended, a design document read or a skip-level with a staff engineer. [exec]
- Keeps an explicit, short docket of executive-owned technical decisions and delegates everything else with real authority. [exec]
- Takes back a docket decision (the differentiating capability, the strategic dependency) that is about to be made by whoever felt strongly. [exec]
- Failure mode: the executive as chief architect. [exec]
- Failure mode: technical judgment outsourced entirely to whoever presented last. [exec]
- Failure mode: dashboards with no contact with reality. [exec]
- Failure mode: discovering strategic lock-in during a renewal negotiation. [exec]
- Failure mode: a docket so long it is micromanagement with a title. [exec]

7. Tensions

Closeness versus bottleneck. The closer a manager stays to the code, the better their judgment, and the more the team waits on them. Reading without approving resolves it. Once reading turns into gatekeeping, the closeness costs more than it buys.

Ownership versus outcome. Letting a team own a decision the manager thinks is wrong puts the outcome at risk. Overriding it puts the team's ability to decide at risk. The threshold of reversibility and cost decides which risk to take, and the failure is applying it according to mood.

Honesty versus authority. Saying "you know this better than I do" can feel like giving up standing, and pretending to understand loses it faster. Credibility at scope is built on questions, and the questions are better when the manager is honest about what they do not know.

Process versus speed. Installing a decision process takes longer than giving the answer, and the answer teaches nothing. For decisions that recur, the process is the more valuable intervention. In a real emergency, the answer is.

Delegation versus strategy. Most architecture should be delegated to senior engineers, and that is mostly what the executive should do. Yet a handful of technical decisions are inseparable from company strategy and cannot be delegated without being made by default. The docket marks the line, and it is kept short on purpose.

8. Worked scenario

A senior manager with four teams learns, in a weekly one-on-one with one of their managers, that a team is about to commit to a new event-streaming platform for a service three other teams consume. The manager supports the choice, and the team's lead engineer is enthusiastic. The platform is capable. Even so, the senior manager, who ran a similar migration on a different technology years ago, believes it is the wrong call. The operational burden will land on a team that has never run stateful infrastructure, and nobody has consulted the consumers.

The pull is to say no. The senior manager knows the domain and has the authority, and the manager would comply. But that would be the nostalgia-driven intervention this unit warns about, since the senior manager's understanding is of a technology that has moved on. It would also teach the manager that decisions of this kind are not theirs to make.

The senior manager applies the threshold first. Is the decision reversible? Not cheaply. A streaming platform underneath a service that three teams consume is a one-way door at department scale, and it quietly commits the whole department to operating it. Does it cross team boundaries with material consequences? Yes, and the consumers have not been asked. The threshold is met on scope and irreversibility, so the senior manager is right to be involved. The judgment lies in the form that involvement takes.

They do not overrule. Working through the manager, they install a decision process. The choice will be made against criteria the senior manager names: operational burden and who carries it, impact on the consumers and their consent, reversibility, and total cost including on-call. The evidence will come from a bounded two-week evaluation, reviewed at the architecture forum. The manager and the lead engineer will decide, with the leads of the consuming teams present. The senior manager is also honest about their own concern and where it comes from, including that their experience was with a different technology.

The evaluation changes the decision. The consumers raise a constraint nobody had considered, and running the platform turns out to need a platform team that does not exist. The manager chooses a managed service instead. The decision is still theirs, and a better one, and the department now has a process it will use again.

The senior manager avoids both of the comfortable options: saying nothing because it was the team's call, and saying no because they once knew this area.

9. Related competencies

- PL-1 Delegation, ownership, and decision rights: the intervention threshold as decision rights, and not taking delegated work back.
- PL-6 Leading through leaders: intervening through a manager instead of around them.
- TJ-1 Evaluating architectural decisions and their organisational consequences: the architectural decisions the docket keeps and the forum reviews.
- SV-4 Engineering economics: build-versus-buy for the differentiating capability, and strategic dependency as the executive's decision.
- CC-2 Coaching versus directing: the coaching form of intervention, in which questions develop judgment and answers do not replace it.

10. Self-check

1. Which kind of technical knowledge persists after you stop coding, and how is it maintained? *Answer:* Systems understanding: the shape of the system, its risks, its dependencies, and who knows what. It is maintained by reading designs and code without approving them, staying in the room for hard decisions, and going deliberately deep in one area each quarter.
2. What two factors most justify stepping into a technical decision? *Answer:* A high cost of being wrong, and difficulty of reversal. Examples are one-way doors, safety, security, data integrity and customer impact in flight. Reversible decisions, where owning the outcome teaches more than the correction would, stay with the team.
3. When you do intervene, how? *Answer:* By raising what was missed, with a question that adds a consideration, and without overriding by authority. If the question does not change the decision, the team owns the outcome.
4. What triggers a manager of managers' direct involvement, and why is installing a process usually better than giving the answer? [org] *Answer:* The triggers are scope across team boundaries with material consequences, irreversibility at department scale, a deadlock that costs more than a decision would, or a decision that quietly commits the whole department. A process (criteria, evidence, decider, date) produces a better decision and leaves the organisation with a capability. An answer produces one decision and teaches managers not to decide.
5. What does "reviewing designs rather than code" preserve, and what map is being maintained? [org] *Answer:* Credibility at one remove. The manager understands every system's shape, risks and dependencies well enough to know where to look harder and

whom to believe, without competing with engineers' depth or becoming an approver. The map is of systems, risks and experts; it is not a claim to mastery.

6. Which technical decisions belong on an executive's personal docket, and what makes fully delegating them a strategy error? [exec] *Answer:* Bet-the-company technology choices, build-versus-buy for the differentiating capability, vendor lock-in on a core capability, infrastructure the product will increasingly stand on, and the security and reliability posture. Delegating them fully does not remove the decision. It means they get made by default, by whoever felt strongly.
7. How does an executive keep aggregate signals honest without becoming the organisation's most senior micromanager? [exec] *Answer:* Structured sensing plus sampled raw signal. A small set of organisation-level trends is checked through direct exposure on a sampling basis (an incident review attended, a design document read, a skip-level with a staff engineer), and principal engineers know the executive wants the truth instead of reassurance.

Sources

- Andrew S. Grove, High Output Management (1983): task-relevant maturity, applied to the manager's own decaying currency.
- Jeff Bezos, 2015 Letter to Shareholders (Amazon.com, Inc.) (2016): one-way and two-way doors as the basis of the intervention threshold.

TJ-5 AI-assisted engineering

adoption, governance, capability-building, and risk

1. Definition and why it matters

AI-assisted engineering is the use of large-model tools in how software is written, reviewed, tested, operated and understood. The competency is the manager's judgment about it: how a team or organisation adopts the assistance, governs its use, builds the capability to use it well, and manages the risks it brings. No v1.0 competency covered this. The previous edition of the standard mentioned AI as an example of a technology that might be transformative but was unproven, and left it there. It now has its own unit for two reasons. It is no longer optional, because engineers use the tools whether or not anyone decided they should. And the evidence about what the tools do is mixed, which makes this the part of the standard where a considered judgment and a fashionable one lie furthest apart. It matters because the tools change what review must establish, what a hiring interview measures, what a productivity metric means and what engineers understand about the systems they own. The same tools can produce an organisation that ships faster or one that no longer understands its own code, depending on decisions a manager makes or fails to make. The examinations test it through a chief executive who asks whether to adopt everywhere, an engineer who ships code they cannot explain, a rollout measured by licences, and a board that expects the tools to cut headcount.

2. Core principles

1. **Decide the adoption; do not let it happen.** What the tools are used for, what they may not touch, and what changes about review, testing and ownership are all decided explicitly, before or during adoption. Nobody should be finding them out after an incident.
2. **Ownership does not move.** Whoever submits a change owns it and can explain it, whatever produced the text. Verification falls to the submitter, and disclosure is required wherever the organisation says so.
3. **Govern proportionately to risk.** Decide what must be verified, which data may be sent to which tools, and what is permitted on which paths. A documentation change and a payment path are governed differently, and blanket prohibition is as much a failure as having no governance.
4. **Build the capability, not just the licence count.** Using the assistance well is a skill. The organisation invests in it and shares what works. A team whose engineers can no

longer explain their own service is an early warning, whatever it does for productivity figures.

5. **Measure in your own conditions and report honestly.** The evidence is mixed and self-reports are unreliable. The effect is measured against outcomes in the organisation's own codebases, and the answer to a board may be "not yet, and here is why".
6. **Size the bet to teach.** A capability that might be transformative but is unproven gets a bounded bet. It is large enough to build capability and evidence, has checkpoints for doubling down or folding, and is funded by explicitly stopping something else instead of being sprinkled across everyone's full plate.

---

3. Models and evidence

The evidence in this unit disagrees with itself. The unit grades that disagreement and does not resolve it by preference.

AI-assisted developer productivity [contested]

These are the two kinds of evidence introduced in DE-5 Designing how work flows. In the controlled experiment in Sida Peng, Eirini Kalliamvakou, Peter Cihon and Mert Demirer, The Impact of AI on Developer Productivity: Evidence from GitHub Copilot (2023), a defined task was completed substantially faster with an assistant. In the randomised trial in Joel Becker, Nate Rush, Elizabeth Barnes and David Rein (METR), Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity (2025), experienced developers working in their own mature repositories were slower with the tools, while believing they were faster. The studies measured different things. The honest management position is that the effect depends on the work, the codebase and the engineer, that perceived speed-up is not a measurement, and that the effect has to be established locally. A manager who cites only the study that supports their preference has failed this unit's central test.

The ironies of automation [practice]

Lisanne Bainbridge, Ironies of Automation (1983) analyses what happens when a task is automated. The human is left with the parts the automation cannot do, which are the hardest. The skills needed to step in when the automation fails decay because nobody practises them any more. And the operator is asked to monitor a system they understand less and less. The paper was written about industrial control, but it describes the risk this unit is most concerned with: an engineering organisation whose engineers can no longer explain, debug or extend the code the tools produced. It is also why capability-building and review for understanding cannot be treated as optional overheads.

Security of AI-generated code [research]

Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt and Ramesh Karri, Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions (2022) found that a substantial share of assistant-generated code, across security-relevant scenarios, contained known classes of vulnerability. The proportion changes with tools and models. The governance consequence does not. Generated code is checked as a class, through scanning and review aimed at the known failure modes, and the paths where a vulnerability would cost most get the most verification. TJ-6 Security, privacy, and compliance as engineering responsibilities covers the security side.

Proportionate AI governance [practice]

This is not a named model. It is the standard this unit proposes: a short, explicit policy stating what the tools may be used for, what data may be sent where, what verification and disclosure each class of change requires, and who owns a change whatever produced it. The policy is scaled to the risk of the path, reviewed as the tools change, and enforced in the pipeline where possible. It can fail in two opposite ways. Blanket prohibition gets ignored, and the absence of any policy gets discovered in an incident.

Staged bets and checkpoints [practice]

The practice from SV-3 Roadmapping, applied to the capability itself. A technology that is transformative and unproven gets a bounded bet, enough to build real capability and learn fast, with explicit checkpoints for doubling down or folding. It is funded by clearing something out, never by adding it to every team's plate. A bet too small to matter is an error, and so is one too big to survive being wrong.

---

4. Practice

The adoption decision, written

One page, written before or at adoption. It says what the tools are for, what they may not touch, what data may be sent to them, what review and testing now require (the checklist in TJ-3 Code review and quality standards), what disclosure is expected, and how the effect will be measured. It is revisited quarterly, because the tools change faster than most policies do.

The understanding check

TECHNICAL JUDGMENT

Every so often an engineer is asked to explain a recent change in their own area, in their own words: how it works, what it assumes and what would break it. This measures the team and is no test of the individual. When the answers start getting thinner, the team's understanding is eroding, and that is the earliest signal this unit knows of.

Capability-building

Time is set aside for engineers to share what works. That includes which tasks the assistance helps with and which it does not, how to verify what it produces, and how to keep the system in your head while using it. The organisation treats fluency with the tools as a skill to develop. A licence on its own does not confer it.

Measuring in local conditions

The rollout is measured against outcomes (cycle time, change failure rate, rework, incident trends) in the organisation's own codebases, over a period long enough to show effects on understanding as well as speed. Self-reports are collected but are not treated as the measure. The report states the uncertainty, and that report is the one that goes upward.

The bounded bet

A capability that could transform the product but is unproven gets a defined scope, a team that owns it, a budget taken from something that was explicitly stopped, and checkpoints with criteria for doubling down or folding. The bet is reviewed on its evidence. Excitement about it counts for nothing.

Governance across teams

At organisational scope the policy is set once, for the seams: data, disclosure, and verification on shared or high-risk paths. Teams choose their own tools inside it. The manager looks across teams for uneven adoption. A team using the tools dangerously and a team that has not started both need attention, for different reasons.

5. Scaling note

At team scope the object is how the team uses the assistance and what review that requires. The manager introduces it with explicit expectations, watches for understanding wearing away, and tells apart the places where it helps from the places where it is used to look fast. At organisational scope the object becomes adoption and governance across teams, with proportionate rules at the seams, capability built deliberately, and uneven adoption noticed and dealt with. It also includes measuring the effect in the department's own conditions. At executive scope the object is the organisation's strategy for AI-assisted engineering.

TECHNICAL JUDGMENT

Investment, governance, capability and risk are treated together as a strategic bet; the executive states honestly what changes about the work and the workforce, evaluates an emerging capability as a bounded bet, and holds the conversation with a board that expects the tools to reduce headcount. How Judgment Scales sets out the pattern; the economics of the AI option are in SV-4 Engineering economics.

6. Judgment

- Introduces AI assistance to the team with explicit expectations for verification, ownership and disclosure. [team]
- Recognises the risks the assistance brings (unverified code, leaked data, eroded understanding) and puts proportionate controls in place. [team]
- Tells apart where the assistance helps the team and where it is being used to look fast. [team]
- Treats an engineer who cannot explain the code they shipped as a signal about the review and the adoption, and does not treat it as a personal failing.
- Failure mode: letting adoption happen without a decision. [team]
- Failure mode: blanket prohibition, which is ignored. [team]
- Failure mode: measuring the adoption by how fast pull requests arrive. [team]
- Failure mode: citing the study that supports your preference and ignoring the one that does not.
- Governs AI-assisted work across teams in proportion to risk: what is permitted, what must be verified and what data may be used, scaled to the risk of the path. [org]
- Builds the capability to use the assistance well across the department, and shares what works. [org]
- Reads uneven adoption across teams and deals with both the dangerous and the absent. [org]
- Recognises a team whose engineers can no longer explain their own service as an early warning that understanding has eroded. [org]
- Measures the effect in the department's own conditions and reports it honestly, including where it is not helping. [org]
- Failure mode: a department-wide rollout measured by licences used. [org]
- Failure mode: governance that prescribes every team's tools instead of setting the rules at the seams. [org]
- Failure mode: treating a productivity gain as real because engineers say it is. [org]
-

TECHNICAL JUDGMENT

Sets the organisation's strategy for AI-assisted engineering (investment, governance, capability, risk) as a strategic bet, and says what changes about the work and the workforce. [exec]

- Evaluates an emerging AI capability that could be transformative and is unproven as a bounded bet with checkpoints, funded by explicit subtraction. [exec]
- Answers a board that expects the tools to reduce headcount with the evidence, the uncertainty and what the organisation knows so far. [exec]
- Reports AI's effect to the board honestly, including a negative or not-yet result. [exec]
- Failure mode: committing the organisation to a headcount plan the evidence does not support. [exec]
- Failure mode: sprinkling the existential capability across everyone's full plate. [exec]
- Failure mode: an AI strategy that amounts to a tools purchase. [exec]
- Failure mode: stampede or prohibition in place of a bounded bet. [exec]

7. Tensions

Speed versus understanding. The tools can raise output while lowering the team's understanding of its own systems, and the second effect shows up later than the first. The judgment is to hold review to understanding, to measure understanding as well as throughput, and to accept that some of the apparent speed was borrowed.

Governance versus adoption. Rules strict enough to remove the risk get ignored, and rules loose enough to be followed leave risk behind. Proportionality resolves it, since the payment path and the documentation change are governed differently, and the rules are enforced wherever they can be, in the pipeline.

Evidence versus urgency. The board wants a return this quarter, and honest measurement takes longer. The judgment is to report the uncertainty instead of the convenient number, having set the review date at the time the investment was made.

Bet versus caution. A capability that could transform the product is also unproven. Too small a bet learns nothing, and too large a bet cannot survive being wrong. The bounded bet with checkpoints resolves it, and the discipline lies in honouring the checkpoints.

Workforce honesty versus reassurance. What the tools change about the work is uncertain, and the people doing the work are anxious about it. Saying what the organisation knows and does not know, including about headcount, is harder than reassuring people. It is also the only version that survives the next quarter.

8. Worked scenario

TECHNICAL JUDGMENT

At a quarterly meeting, the board asks a head of engineering to present a plan for cutting engineering headcount by a third over two years on the strength of AI assistance. The chair cites a vendor's productivity figures and a competitor's public statement. The chief executive is sympathetic to the board and wants the plan within a month. The organisation adopted assistants a year ago. Adoption is uneven and the measured effects are mixed, and two teams have had incidents traced to generated code nobody understood.

The two easy responses are to produce the plan, because the board asked and the chief executive wants it, or to refuse, because the evidence does not support it. The first commits the organisation to a headcount trajectory built on a vendor's number. The second reads as engineering resisting the future, and it is not even the honest answer, which is more specific than no.

The head of engineering brings the evidence, framed in the board's terms. The vendor's figure comes from a controlled experiment on a defined task. The strongest independent trial found experienced developers slower in real codebases while believing they were faster. Over a year, the organisation's own measurements show cycle-time gains on some kinds of work and none on others, two incidents from unverified generated code, and an erosion of understanding in two teams that took a quarter to repair. The head of engineering states plainly what the organisation knows. The tools help with some work. Their effect does not yet amount to a headcount figure. A plan built on a third would rest on a number nobody has measured here.

They then offer something the board can act on: a strategy for AI-assisted engineering as a bounded, staged bet. It has governance at the seams, capability built deliberately across teams, a defined measurement of the effect on outcomes over the next three quarters, and two checkpoints at which the board will see the organisation's own numbers. They describe what changes about the work, including review, testing and the shape of some roles, and say that any change to headcount will follow the measured effect and will not run ahead of it. They add that if the effect turns out to be large, the organisation should be the first to know, and the board will hear it from them.

The board is not fully satisfied, and the chair says so. The chief executive backs the staged plan, because the organisation can be held to it. The head of engineering does not put in front of the board a headcount number the evidence did not support. That would have been the easiest thing to say, and the organisation would then have spent two years failing to deliver it.

9. Related competencies

- TJ-3 Code review and quality standards: what review must establish for generated code, and the standards floor.

- DE-5 Designing how work flows: AI assistance as part of how work flows, where the contested evidence is introduced.
- SV-5 Engineering metrics and measurement: measuring AI's impact honestly and reporting it upward.
- SV-4 Engineering economics: the AI service as a third option beside build and buy.
- TJ-6 Security, privacy, and compliance as engineering responsibilities: the security profile of generated code, and the data sent to tools.
- PL-3 Hiring and onboarding: assessing capability in hiring, as opposed to access to tools.

10. Self-check

1. What is the honest management position on the evidence for AI-assisted productivity? *Answer:* That it is mixed. A controlled experiment on a defined task found a large speed-up, and a randomised trial of experienced developers in their own repositories found a slowdown they did not perceive. The effect depends on the work, the codebase and the engineer; perceived speed-up is not a measurement; and the effect has to be established locally.
2. What do the ironies of automation predict for an engineering organisation, and what practice answers them? *Answer:* Engineers will be left with the hardest parts, the skills to step in when the automation fails will decay, and they will be asked to own systems they understand less and less. Review for understanding, the understanding check and deliberate capability-building are the answer.
3. An engineer ships code generated by an assistant that they cannot explain in review. What is the signal, and what is not? *Answer:* It is a signal about the review practice and the adoption decision, since review is passing changes it should stop. It is not primarily a personal failing, because the system passed it. The fix is review that establishes understanding, with ownership that stays where it was.
4. What does proportionate governance of AI-assisted work across teams contain, and what are its two failures? [org] *Answer:* What is permitted, what must be verified, what data may be sent where, and disclosure, all scaled to the risk of the path and set at the seams, with tools left to the teams. The failures are blanket prohibition, which is ignored, and no policy at all, which is discovered in an incident.
5. How is a department-wide rollout measured? [org] *Answer:* By what teams can now do, at a quality they can defend, against outcomes in their own conditions: cycle time, change failure rate, rework and understanding. Licences used and self-reported speed are not the measure.
6.

The board expects AI to reduce engineering headcount and wants a plan. What is the answer? [exec] *Answer:* The evidence, in the board's terms and including the organisation's own measurements, and a staged, bounded strategy with checkpoints at which the board sees local numbers. Any headcount change follows the measured effect instead of preceding it, and the organisation will be the first to say so if the effect is large.

7. How is an emerging AI capability that could transform the product but is unproven evaluated? [exec] *Answer:* As a bounded bet sized to teach, large enough to build real capability and evidence, with checkpoints for doubling down or folding, and funded by explicitly clearing something out. It is neither prohibited nor stampeded into.

Sources

- Sida Peng, Eirini Kalliamvakou, Peter Cihon and Mert Demirer, The Impact of AI on Developer Productivity: Evidence from GitHub Copilot (2023): the controlled experiment on AI-assisted task completion.
- Joel Becker, Nate Rush, Elizabeth Barnes and David Rein (METR), Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity (2025): the randomised trial on experienced developers with AI tools.
- Lisanne Bainbridge, Ironies of Automation (1983): the ironies of automation, covering what automation leaves to the human and what it erodes.
- Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt and Ramesh Karri, Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions (2022): the vulnerability profile of AI-generated code.

TJ-6 Security, privacy, and compliance as engineering responsibilities

owned by engineering, not delegated to a function

1. Definition and why it matters

Security, privacy and compliance are engineering responsibilities. Engineers build and operate the systems that hold customers' data, the code that exposes or protects it, and the means of meeting the obligations the organisation has taken on, and no security function can own what engineering does not. v1.0 left all three implicit inside quality and risk. v3.0 gives them a competency of their own because they involve their own judgment calls, such as triaging a scanner's four hundred findings, sizing a compliance request against a delivery quarter, or deciding what data may be sent to which tools, and because the ways they fail are distinct and expensive. The competency does not require being a security engineer. It requires treating a vulnerability, a privacy obligation or a compliance requirement as the team's own work, prioritised by risk instead of being deferred to a function or frozen in fear. It also means assigning ownership for gaps that span teams, weighing a mandate honestly against delivery and, at scope, owning all of this as organisational risk at executive level. It matters because getting it wrong costs customers' trust and has legal consequences. And the most common failure is diffusion, not negligence: everyone assumed another team owned it. The examinations test it through a team paralysed by a scanner, an engineer with unaudited production access, a "two-week" compliance request, a gap that spans four teams, and a mandate that requires re-architecting the organisation.

2. Core principles

1. **Engineering owns it.** A security finding, privacy obligation or compliance requirement in a team's system is that team's work, prioritised alongside everything else. A security function advises, audits and supports. It cannot own what it does not build.
2. **Triage by exploitability and exposure, not by count.** Four hundred findings are not four hundred equal risks. Prioritise by what is reachable, what it exposes and how large the blast radius is, and where a class exists, fix the class.
3. **Privacy and access are everyday decisions.** Who can see production data, where copies of personal data live and what gets logged are all settled in ordinary engineering choices, long before they become incidents.
4. **Assign an owner to a gap that spans teams, and fix the pattern.** A finding that spans several teams with no owner needs a named one and a class-level fix. The fix goes into whatever produces the pattern, such as the pipeline, not into today's instances.

5. **Weigh a mandate honestly against delivery.** Separate what is mandatory from what is merely requested, size it against the quarter, and take the trade-off to the level that owns the priority. Refusing it is wrong, and so is absorbing it silently.
6. **At scope, it is organisational risk owned at the executive table.** Investment, accountability and exposure are stated in business terms. A mandate that requires re-architecting systems is led as a programme and sequenced by risk, whoever happens to be shouting loudest.

---

3. Models and evidence

Security engineering has a large practitioner literature and a growing research one. The unit draws on both and grades each.

Threat modelling [practice]

Adam Shostack, Threat Modeling: Designing for Security (2014) gives this discipline its standard treatment. It asks four things of a system: what are we building, what can go wrong, what are we going to do about it, and did we do a good job? Its value to managers is that it makes security part of design instead of a scan at the end, and that a manager can ask its questions in a design review without being a security engineer. Its limit is that a threat model is a snapshot. Systems and threats move on, and a model nobody revisits describes a system that no longer exists.

Risk-based triage [practice]

Not a named model, but the practice this unit offers as the answer to the paralysed team. Findings are prioritised by exploitability (is the vulnerable path reachable?) and by exposure (what would it give an attacker, and how many people are affected?), and where several findings share a cause, the class is fixed. Its provenance is ordinary security practice. Its failure modes are the two the examinations present: freezing at the count, and ignoring it.

Privacy by design [practice]

Ann Cavoukian, Privacy by Design: The 7 Foundational Principles (2009) sets out the principles: privacy as the default, built into design instead of bolted on, with data minimised and its handling visible. For engineering managers the practical content lies in everyday decisions about what is collected, where it is copied, who can read it, what is logged and how deletion works across systems in practice. It also means recognising that when a deletion request finds personal data duplicated across four systems, two backups and an analytics warehouse, that is a design failure, made one decision at a time.

Security of AI-generated code [research]

Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt and Ramesh Karri, Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions (2022) found that generated code carries known classes of vulnerability at a substantial rate. For this unit, the consequence is that generated code is scanned and reviewed for those classes on the paths where they matter, and the data sent to AI tools is governed like any other data flow. Neither point argues against the tools. Both argue for treating them as part of the system's attack surface.

Compliance as a programme [practice]

Also not a named model. This is the executive-scope practice for a security or compliance mandate that touches many systems. It has ownership, sequencing by risk, explicit capacity, an honest roadmap impact and reporting, and it is led like the strategic effort it is. It fails when the mandate is squeezed invisibly into team slack, which is how mandates get missed and how the organisation learns it was never serious about them.

---

4. Practice

The finding triage

When a scanner or an audit produces a list, the team sorts it before acting. Is each finding reachable? What does it expose, and to whom? Do several findings share a cause? The result is a short list of classes to fix, in order of risk. Unreachable findings are recorded and deferred; they are neither fixed first nor ignored.

The access review

Every quarter, the team lists who can read or write production data and why. Standing write access without an audit trail is removed in the week it is found, without waiting for the next review. The problem was the grant, and the engineer who held it is not to blame.

Threat modelling in the design review

For any system that handles personal or financial data, the design review includes the four questions: what are we building, what can go wrong, what will we do about it, and did we do a good job? The manager asks them, the engineers answer them, and the answers are kept with the design.

The mandate sizing

TECHNICAL JUDGMENT

When a compliance request arrives, the team sizes it honestly: what is mandatory, what is requested, and what it will take. The manager takes the result, with its impact on delivery and the options, to the level that owns the priority. The request is neither refused nor absorbed.

The cross-team owner

When a finding spans several teams, the manager of managers names an owner for the class. They fund a fix to the pattern itself (the shared library, the pipeline check, the template) and track it to completion in every team. Instances fixed one at a time grow back.

The exposure register

At executive scope, the organisation keeps a short register of its security, privacy and compliance exposure in business terms: what is most likely, what would cost most, and what has no owner. It is reviewed alongside the risks the executive tracks personally, and it is the basis of the investment case at the executive table.

---

5. Scaling note

At team scope the object is the team's own security practice and any vulnerability in its service. The manager treats that work as the team's, triages by risk, and catches privacy and access problems in everyday decisions. At organisational scope the object becomes a class of gap across several teams, where ownership is assigned and the pattern fixed. It also covers weighing a compliance requirement honestly against delivery and separating the mandatory from the requested. At executive scope, security, privacy and compliance are organisational risk owned at executive level. Investment and accountability are stated in business terms. A mandate that requires re-architecting systems is led as a programme sequenced by risk, and a serious vulnerability class across many services is prioritised as exposure, with explicit capacity, instead of being squeezed into slack. How Judgment Scales sets out the pattern; the incident structure a security incident follows is in DE-3 Reliability, incident response, and operational ownership.

---

6. Judgment

- Treats a security finding, privacy obligation or compliance requirement in the team's system as the team's own work, prioritised by risk. [team]
- Triages a scanner's findings by exploitability and exposure and fixes the class, without either freezing or ignoring them. [team]
-

TECHNICAL JUDGMENT

Spots a privacy or access-control problem in an everyday decision, such as standing production access, personal data copied where it should not be, or sensitive values in logs. [team]

- Asks the threat-modelling questions in a design review without being a security engineer. [team]
- Governs generated code and the data sent to AI tools as part of the attack surface.
- Failure mode: deferring security to a function that does not build the system. [team]
- Failure mode: paralysis at the count. [team]
- Failure mode: ignoring the count. [team]
- Failure mode: discovering unaudited production access in an incident instead of a review. [team]
- Assigns a named owner and a class-level fix when the same gap appears across several teams. [org]
- Weighs a compliance requirement honestly against delivery, separating mandatory from requested and sizing it against the quarter, and takes the trade-off to the level that owns the priority. [org]
- Handles a privacy obligation that touches four teams' systems by establishing who owns each part before assuming another team does. [org]
- Puts security baselines into the standards floor at the boundaries, and leaves the tools inside teams to the teams. [org]
- Failure mode: patching a cross-team finding one instance at a time. [org]
- Failure mode: accepting "two weeks" from legal for a quarter's work, or refusing the mandate. [org]
- Failure mode: every team assuming another owns the obligation. [org]
- Owns security, privacy and compliance as organisational risk at executive level, stating investment, accountability and exposure in business terms. [exec]
- Leads a mandate that requires re-architecting systems across the organisation as a programme, with ownership, sequencing by risk, capacity and reporting. [exec]
- Prioritises a serious vulnerability class across many services as business exposure, with explicit capacity and an honest roadmap impact. [exec]
- Keeps security posture on the short list of risks tracked personally. [exec]
- Failure mode: treating security capacity as something teams find down the back of the sofa. [exec]
- Failure mode: a mandate squeezed invisibly into team slack until it is missed. [exec]
- Failure mode: learning the organisation's exposure from the breach. [exec]

7. Tensions

Risk versus delivery. Every hour spent on a finding is an hour not spent on features, and every finding left open is exposure. Triage by exploitability and exposure turns the trade-off into a decision, where otherwise it would be a reflex in one direction or the other.

Ownership versus expertise. Engineering owns security without having a security engineer's depth. The security function advises, audits and builds the shared tooling, and the team owns the work in its own system. Either side can fail: the team that defers everything, or the function that owns nothing it can change.

Access versus friction. Engineers need production access to operate their systems, and standing access is the most common privacy failure. The judgment lies in access that is scoped, audited and time-bounded, inconvenient enough that using it is deliberate and available enough that nobody works around it.

Mandate versus quarter. A compliance requirement is often non-negotiable in substance and negotiable in scope and timing. Sizing it honestly and separating the mandatory from the requested is what lets the manager avoid both refusing it and absorbing it.

Programme versus slack. It is tempting to spread a mandate that touches every system as a small tax on every team, and that is how it gets missed. Leading it as a programme costs visible capacity, and it is the only version that finishes.

---

8. Worked scenario

A senior manager with five teams receives the results of an external security audit. One class of gap appears in eleven services across four of the five teams: services that accept unvalidated input on internal interfaces, on the assumption that internal callers can be trusted. The audit lists these as eleven separate findings. When asked, each team says the pattern came from a shared service template written three years ago by a team that no longer exists, and each assumes that fixing the template is someone else's job. The security function has filed the findings and asked for remediation dates.

The two easy answers both fail. Asking each team to fix its own findings fixes eleven instances and leaves the template free to produce a twelfth. Escalating the whole thing to the security function hands it to a group that owns none of the services and cannot change them.

The senior manager treats it as a class with no owner. They name one, the platform team lead, who has the closest claim to the template and to the shared library it produces. They then fund a fix to the pattern. It has three parts: an input-validation layer in the shared library, a pipeline check that fails any service without it, and a migration of the eleven services in order of exposure, starting with the two internet-adjacent ones. The ordering is triage.

A finding in the service that handles payment callbacks is not the same risk as one in an internal reporting job, and the audit's list did not make that distinction.

The four affected teams keep their own instances in the migration plan, with dates. The dates follow the shared fix, because a team that patches its instance by hand this week has done work the library will redo. The security function receives the plan and a reporting cadence, and is asked for the one thing it can do that the teams cannot: a targeted re-test of the two highest-exposure services when the migration lands.

The senior manager also asks the question the audit raised and left unanswered. What else did that template produce, and where else does the organisation assume that internal callers can be trusted? The answer becomes the next item on the exposure register.

The senior manager does not treat eleven findings as eleven tasks, and does not treat security as the security function's problem. Either would be how the twelfth finding arrives.

9. Related competencies

- TJ-2 Technical debt: security exposure as a sibling of technical debt, prioritised the same way.
- TJ-3 Code review and quality standards: the security baseline in the standards floor, and generated code checked as a class.
- TJ-5 AI-assisted engineering: data sent to AI tools, and the vulnerability profile of generated code.
- DE-3 Reliability, incident response, and operational ownership: the incident structure a security incident follows (restore, communicate, diagnose).
- DE-1 Prioritisation under constraint: where a mandate goes in the queue, and the cost of saying yes.
- CC-5 Values, integrity, and ethical judgment under pressure: making the costly right call when a compliance requirement is inconvenient.

10. Self-check

1. A scanner reports four hundred dependency vulnerabilities and the team is paralysed. What is the right move? *Answer:* Triage by exploitability and exposure: which vulnerable paths are reachable, what they expose and to whom, and which findings share a cause. Fix the classes in order of risk, and record and defer the unreachable ones. Do not freeze at the count, and do not ignore it.
2. An engineer has had unaudited write access to production data for months. What is the problem? *Answer:* The grant, and not the engineer. Standing, unaudited access is

removed in the week it is found and replaced with access that is scoped, audited and time-bounded. The quarterly access review is what should have caught it.

3. What are the four threat-modelling questions a manager can ask in a design review? *Answer:* What are we building? What can go wrong? What are we going to do about it? Did we do a good job? They make security part of design, and asking them does not take a security engineer.
4. A "two-week" compliance request arrives; legal calls it mandatory; your sizing says a quarter. What do you do? [org] *Answer:* Size it honestly, separate what is mandatory from what is requested, and take the trade-off, with its delivery impact and options, to the level that owns the priority. Do not refuse it, and do not silently absorb it into the quarter.
5. A security audit finds the same class of gap across several teams and nobody owns the fix. What is the fix? [org] *Answer:* A named owner for the class, and a fix to the pattern itself (the shared library, the pipeline check, the template), migrated in order of exposure and tracked across every team. Instances fixed one at a time grow back from the same source.
6. Why is a compliance mandate that touches every system led as a programme rather than distributed as a small tax on each team? [exec] *Answer:* A mandate squeezed invisibly into team slack is how mandates get missed, and how the organisation learns it was never serious. A programme has ownership, sequencing by risk, explicit capacity and reporting, and it finishes.
7. What does an executive own about security that cannot be delegated to a function? [exec] *Answer:* The organisation's exposure as business risk (investment, accountability, what is most likely and what would cost most), stated at the executive table and kept on the short list of risks the executive tracks personally. The function advises and audits, and the exposure stays with the executive.

Sources

- Adam Shostack, Threat Modeling: Designing for Security (2014): threat modelling as a design discipline, and the four questions.
- Ann Cavoukian, Privacy by Design: The 7 Foundational Principles (2009): privacy by design.
- Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt and Ramesh Karri, Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions (2022): the vulnerability profile of AI-generated code.

Culture & Coaching

What this domain is

Culture & Coaching is the work of shaping the environment in which good engineering happens and developing the judgment of the people in it. v3.0 adds a third part: sustaining the manager who does both. Culture is the accumulated pattern of which behaviour gets rewarded, which gets tolerated and which gets punished; a values poster is no part of it. Coaching is how a leader makes people stronger instead of more dependent. The domain carries 20% of every examination: 12 questions on EMA-I, 15 on EMP-II and 18 on EME-III. Its six competencies are listed below with the questions each carries at each level, and the Exam Content Outlines give the proficiency indicators.

The throughline

In every competency here the leader is shaping an environment. That means building safety without lowering standards, developing people instead of rescuing them, distributing opportunity fairly, sustaining motivation and pace, modelling values most visibly when they cost something, and managing their own state, because the team copies it. The domain has the strongest research behind it: psychological safety, intrinsic motivation and burnout are among the best-evidenced findings this standard cites. It is also where the comfortable answer is most reliably wrong. The reassurance comes in place of the truth, the rescue in place of the question, the reminder in place of the redesign; the crunch is praised, the value is waived for the launch, and the manager absorbs evenings until they break.

How the domain scales

At team scope you shape one team's environment directly, through your reactions, your consistency, your coaching and your own boundaries. At organisational scope you shape culture through your managers. You coach the coaches, read safety from signals, fix the dynamics between teams at their causes, engineer equity across sites, and correct overwork while the metrics still look good. At executive scope culture turns out to be the most consequential system you run. It is shaped by what leadership visibly does, survives growth, reorganisations and mergers only if someone leads it deliberately, underpins the organisation's trust in you, and copies your own state. The general pattern is in How Judgment Scales; each unit's scaling note places its own competency on it.

Reading order

Read CC-1 Psychological safety and productive conflict first, since safety is the precondition for every honest signal the rest of the standard depends on. CC-2 Coaching versus directing comes next, because coaching is the mechanism through which most of the people competencies are exercised. CC-5 Values, integrity, and ethical judgment under pressure is the domain's centre of gravity and belongs alongside CC-1 Psychological safety and productive conflict: what leadership does when a value is costly is where safety and trust are decided. CC-3 Inclusive practices and equitable opportunity distribution and CC-4 Motivation, recognition, and sustainable pace form a pair about the conditions people work in, and both lean on the Distributed and Hybrid Work chapter. Leave CC-6 Self-management until last. It is about the reader.

Across domains

Culture draws its instruments from People Leadership: PL-2 Feedback and difficult conversations for feedback, PL-4 Performance management for the consequences that make values real, and PL-6 Leading through leaders for the managers through whom culture is shaped at scope. From Delivery & Execution it takes its most consequential ritual, DE-3 Reliability, incident response, and operational ownership's blameless review, and its most legible test, the error budget obeyed when it is inconvenient. The structural half of a merger is SV-6 Organisational design and leading change. The intervention threshold that stops coaching turning into rescue belongs to TJ-4 Technical credibility and intervention.

CC-1 Psychological safety and productive conflict

safe to speak, expected to disagree, committed once decided

1. Definition and why it matters

Psychological safety is a shared belief that the environment is safe for interpersonal risk-taking: admitting you are stuck, asking the naive question, challenging a design, flagging an unrealistic deadline, raising a problem early. The evidence puts it among the strongest predictors of team performance there is. It is also the precondition for every honest signal this standard relies on, including bad news, dissent, blameless reviews and upward feedback. Its twin is productive conflict, meaning open disagreement about ideas followed by commitment once the decision is made. At the level of the organisation it takes the form of trust in leadership, which behaves the same way: it is built in visible moments, spent by inconsistency and repaid only by behaviour over time. The competency matters because safety is neither niceness nor the opposite of high standards. The aim is to have both, and the commonest misreading is that safety means going easy. Safety is built and destroyed in small moments, and one public humiliation teaches everyone to keep quiet for a year. The examinations test it through the team that never challenges anything, the manager whose team has gone silent, the survey that flatters an unsafe organisation, and the review after a high-profile failure involving a senior person. That review is where the culture gets written.

2. Core principles

1. **Respond to interpersonal risk with curiosity, not punishment.** How the manager reacts to an admitted mistake, a naive question or a challenge is the entire mechanism. Curiosity keeps the channel open, and one public humiliation shuts it for everyone who saw it.
2. **Safety and standards are independent axes, and the goal is both.** Safety without standards produces comfort; standards without safety produce silence and hidden failure. The target is high standards that people can survive.
3. **Invite disagreement about ideas, then expect commitment.** Productive conflict means open argument before the decision and unity after it. Silence is not harmony, and a consensus nobody disagreed with has usually gone unchallenged rather than been agreed.
4. **Read safety from behaviour, not from surveys alone.** The dependable signal is whether people raise bad news early and challenge decisions upward without

ceremony. Where safety is low, even anonymous instruments overstate it, because people distrust the anonymity as well.

5. **Blamelessness is for honest error; accountability is for patterns.** People need to know which of the two worlds they are in. Safety that excuses recklessness does not deserve the name, and neither does accountability that punishes error.
6. **Trust is a behavioural account.** Deposits are made by telling the truth consistently when it costs something, following through visibly, admitting mistakes and explaining decisions. Reassurance draws on the account and only behaviour pays into it.

---

3. Models and evidence

No competency in the standard has better evidence behind it, and the unit grades its central claim to match.

Psychological safety [research]

Amy C. Edmondson, Psychological Safety and Learning Behavior in Work Teams (1999) defined and measured the construct as a shared belief held by members of a team that the team is safe for interpersonal risk-taking. That study, and the literature that followed it, found that it predicts team learning behaviour and performance. The book-length treatment in Amy C. Edmondson, The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth (2018) adds the point this unit insists on. Safety and standards vary independently: high safety with high standards is the learning zone, and high safety with low standards is the comfort zone. Safety does not mean niceness or comfort, and it does not mean an absence of conflict.

The five keys of team effectiveness [research]

Google studied what distinguished its effective teams, as reported in Julia Rozovsky, The Five Keys to a Successful Google Team (2015), and found psychological safety the strongest of five differentiators, ahead of dependability, structure and clarity, meaning, and impact. It is a single-company study and its value is confirmatory. In a large engineering organisation, using its own data, the construct from the academic literature turned out to matter most.

Conflict then commitment [practice]

Patrick Lencioni, The Five Dysfunctions of a Team: A Leadership Fable (2002) places trust at the base of a team's functioning, with productive conflict as what trust makes possible and commitment as what conflict produces. The matching failures are artificial harmony, where conflict is avoided, and ambiguity, where commitment was never reached. It is a

practitioner model and is graded as one. This unit uses it for vocabulary for the second half of the competency: disagreement is meant to happen before the decision and to stop after it.

Blameless postmortems [practice]

This is the review practice from DE-3 Reliability, incident response, and operational ownership, treated here as the most important ritual of safety. A review that asks how the system allowed the failure is how honesty about failure becomes routine instead of depending on charisma. It carries most cultural weight when the stakes are highest and a senior person is involved, because that review shows everyone whether blamelessness applies above a certain pay grade.

Trust as a behavioural account [practice]

This is the working description of trust in leadership used in this unit, and is not a named model. Trust is an account that only behaviour pays into: telling the truth consistently, especially when it is costly; following through visibly on commitments; admitting mistakes; explaining decisions, including the ones people hate. Inconsistency, spin and broken promises draw it down. It follows that a reassurance campaign in response to falling trust is a withdrawal dressed as a deposit.

---

4. Practice

The reaction, rehearsed

The manager decides beforehand how to respond to the next admitted mistake, naive question or challenge to their own idea. The moment will come without warning, and the team reads the reflex. Phrases such as "Thank you for telling me; what do you need?" and "good catch, I was wrong" get practised until they come automatically.

Dissent, then a decision

Significant decisions are made so that disagreement comes first. The manager asks the quietest person directly and looks for the strongest objection before deciding. Then the decision is closed explicitly and the expectation of commitment is stated. A decision nobody argued with gets revisited. When everyone goes on arguing with a decision after it was made, that is a commitment failure, and the manager calls it one.

Reading safety through behaviour

In place of a score, the manager tracks two behavioural signals: how early bad news arrives, and how often decisions are challenged upward. In skip-levels and meetings they listen to what people say. A decent survey score in a quiet organisation is a hypothesis to test, and the manager does not treat it as a finding.

The two worlds, explained

The team is told once, clearly, which behaviours meet with curiosity (honest error, admitted ignorance, dissent) and which are handled as performance matters (patterns of recklessness, of unaccountability, of failing to do what was agreed). Safety and accountability complement each other, and this explanation stops people mistaking one for the other.

Honest uncertainty

When people ask something the manager cannot answer, such as "are we next?" after layoffs elsewhere, the reply covers what is known, what is not, and what the manager will do once they learn more. Manufactured reassurance buys comfort tonight with credibility tomorrow, and people remember it.

5. Scaling note

At team scope the object is safety in the room. The manager's own reactions to bad news and dissent build or destroy it, and it is paired with high standards. At organisational scope the manager mostly reads safety through signals and builds it through other managers. That means coaching a manager whose team has gone quiet on that manager's own behaviour, turning personal safety into structural safety through taught norms and rituals, and dealing with conflict between teams at its causes through shared outcomes and a single joint review. The manager also has to pass the personal trust test of honest uncertainty. At executive scope safety is a property of the organisation: leaders are hired and promoted for it, it is measured honestly, and it is protected when pressure is greatest. Trust in leadership is the executive's currency. Hard decisions carried out with visible integrity build it, and spin spends it. The pattern is set out in How Judgment Scales; what leadership visibly does when a value is costly is covered in CC-5 Values, integrity, and ethical judgment under pressure.

6. Judgment

- Responds to an admitted mistake, a naive question or a challenge with curiosity. [team]
- Pairs safety with high standards and can explain why "going easy" is not safety. [team]
- Invites open disagreement about ideas and then expects commitment. [team]

CULTURE & COACHING

- Models fallibility: "good catch, I was wrong." [team]
- Failure mode: reading safety as going easy. [team]
- Failure mode: punishing the messenger. [team]
- Failure mode: mistaking silence for harmony. [team]
- Failure mode: the one public humiliation that silences a team for a year. [team]
- Reads safety from behaviour (bad news early, decisions challenged upward) and checks surveys against skip-levels and what people say. [org]
- Coaches a manager whose team never speaks up on that manager's own behaviour: how they meet challenge, whether they model uncertainty, whether dissent has ever been rewarded in their room. [org]
- Turns personal safety into structural safety as the organisation grows, with norms taught, leaders selected partly on how they handle bad news, and rituals that make it routine. [org]
- Fixes a rivalry between teams at its causes, such as competing goals and scarce recognition, with shared outcomes and joint work. [org]
- Runs one joint, blameless review when two teams blame each other after a shared incident, because separate reviews harden separate stories. [org]
- Answers "are we next?" with what is known, what is not, and what will happen when more is known. [org]
- Failure mode: taking a decent survey score at face value in a quiet organisation. [org]
- Failure mode: fixing rivalry with team-building events. [org]
- Failure mode: letting teams keep separate versions of the truth after an incident. [org]
- Failure mode: promising what cannot be known. [org]
- Makes safety a property of the organisation: hires and promotes leaders who build it, measures it honestly, protects it when pressure is greatest. [exec]
- Runs the review after a high-profile failure involving a senior person by the same rules as any other, with a system focus, honest analysis and senior accountability for systemic causes. [exec]
- Explains accountability and safety as complements, so people know which world they are in. [exec]
- Treats falling trust in an engagement survey as data about leadership behaviour: acknowledges it publicly, investigates the causes and responds with visible change. [exec]
- Is present and speaks plainly when the story about a reorganisation turns hostile; owns the costs, explains the reasoning again, stays available. [exec]

- Carries out painful decisions with integrity: truthful about the reasons, humane in how it is done, generous to those leaving, honest with those staying. [exec]
- Failure mode: blameless until senior. [exec]
- Failure mode: a reassurance campaign in place of behaviour change. [exec]
- Failure mode: disputing the survey. [exec]
- Failure mode: spin, on which distrust compounds. [exec]
- Failure mode: leading a reorganisation from behind a memo. [exec]

7. Tensions

Safety versus standards. The temptation is to trade one for the other, either relaxing standards to make people comfortable or tightening them and accepting silence as the price. The two vary independently. The tension only feels real to a manager who has never seen both held at once.

Dissent versus commitment. A team that argues well before a decision may go on arguing after it, and a team that commits cleanly may stop arguing too early. The resolution is an explicit close ("we have decided; now we commit"), with the manager saying which phase the team is in.

Blamelessness versus accountability. A review that names nobody can look like one that holds nobody responsible, and a manager who holds people to account can look like one who blames. Both can exist once they are separated: the review is about the system, and patterns of behaviour are a performance matter dealt with elsewhere.

Measurement versus truth. The organisation wants a safety score, and the score is least reliable precisely where safety is lowest. Behavioural signals and triangulation resolve it. The discipline lies in refusing to be reassured by the number.

Reassurance versus honesty. People in an anxious organisation want to hear that it will be fine. A manager who says so without knowing has spent trust they will need later. Honest uncertainty is harder to deliver, and it is the only version that builds over time.

8. Worked scenario

A head of engineering leads an organisation whose flagship launch failed publicly, at scale, two days ago. Customers and the press know. The postmortem is tomorrow. The decision that contributed most was taken by a vice-president who, under schedule pressure, chose to skip a load test the platform team had recommended. At this morning's executive meeting

the chief executive said that "someone needs to be held accountable", and the board likes the vice-president. Everyone in the organisation knows who made the call.

Both easy responses would be fatal to the culture. A blame exercise, whether it names the vice-president or lets the chief executive's demand set the tone of the room, tells every engineer that blamelessness stops at a certain pay grade, and therefore that it does not exist. A soft review that steers round the decision because of who made it sends the same message from the other direction: senior people are protected and honest analysis is for everyone else.

The head of engineering runs the review by the same rules as any other and announces that in advance. It examines the timeline and the system. Why could one decision skip a recommended load test? Why was there no gate for a launch at this scale? Why did the platform team's recommendation carry no weight? What did executive pressure on the schedule contribute, including the head of engineering's own part in setting it? The vice-president's decision appears in the timeline because it is a fact, and it is examined as a decision taken under conditions the system created. The vice-president attends and answers the same questions anyone else would. The review ends with owned actions: a launch gate with authority, a load-test requirement for changes above a threshold, and a change to how schedule pressure reaches launch decisions.

Accountability is dealt with separately. In private, the head of engineering talks to the vice-president about the judgment involved in skipping a recommended test under pressure. It is framed as feedback on how they lead, with an explicit expectation for the next launch. If a pattern emerges, it becomes a performance matter through PL-4 Performance management. With the chief executive, the head of engineering explains the distinction and what ignoring it would cost. A public sacrifice would meet the demand today and teach the organisation to hide the next problem, which is how the next failure gets worse. The senior accountability the review does establish is for systemic causes, and the head of engineering takes their own share of it in the review, in front of everyone.

The one thing the head of engineering will not allow is for the most closely watched review of the year to be where blamelessness ended.

9. Related competencies

- PL-2 Feedback and difficult conversations: the reaction to bad news that decides whether feedback, upward feedback above all, gets given at all.
- DE-3 Reliability, incident response, and operational ownership: the blameless postmortem as the ritual with most consequence for safety.
- CC-5 Values, integrity, and ethical judgment under pressure: what leadership visibly does when a value is costly, and consequences that reach the powerful.

- CC-2 Coaching versus directing: the coaching stance that develops judgment in place of punishing error.
- PL-6 Leading through leaders: coaching a manager on the behaviour that builds trust in their own room.

10. Self-check

1. How is psychological safety built or destroyed in everyday moments? *Answer:* Through the manager's reaction to an admitted mistake, a naive question or a challenge to their idea. Curiosity keeps the channel open. One public humiliation closes it, for a long time, for everyone who saw it.
2. Why is safety not the opposite of high standards? *Answer:* They vary independently. High safety with low standards gives comfort, and low safety with high standards gives silence and hidden failure. The goal is both: high standards that people can survive.
3. What distinguishes productive conflict from its two failures? *Answer:* Open disagreement about ideas before the decision, and commitment after it. One failure is artificial harmony, where conflict is avoided and silence passes for agreement. The other is ambiguity, where the decision was never closed and the argument carries on.
4. Why do anonymous surveys overstate safety exactly when it is lowest, and what do you triangulate with? [org] *Answer:* In an unsafe organisation people do not trust the anonymity either. Triangulate with behaviour (how early bad news arrives, how often decisions are challenged upward), with skip-levels and with what people say in meetings.
5. Two teams are blaming each other after a shared incident. What is the review? [org] *Answer:* One joint, blameless review, with the fixes owned jointly. Separate reviews harden separate stories.
6. A senior person is involved in a high-profile failure. Why does this one review set the culture more than any policy? [exec] *Answer:* The whole organisation watches it to find out whether blamelessness applies above a certain pay grade, and if it does not, it does not exist. Running it by the usual rules (system focus, honest analysis, senior accountability for systemic causes) is worth a hundred policy documents.
7. An engagement survey shows declining trust in senior leadership. What is the response, and what confirms the diagnosis? [exec] *Answer:* Treat it as data about leadership behaviour. Acknowledge it publicly, look honestly into the causes and respond with visible change. A defensive rebuttal or a reassurance campaign confirms the diagnosis.

Sources

- Amy C. Edmondson, Psychological Safety and Learning Behavior in Work Teams (1999): psychological safety defined and measured, and its link to team learning.
- Amy C. Edmondson, The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth (2018): safety and standards as independent axes; the learning zone.
- Julia Rozovsky, The Five Keys to a Successful Google Team (2015): Google's internal study of team effectiveness.
- Patrick Lencioni, The Five Dysfunctions of a Team: A Leadership Fable (2002): trust, conflict and commitment as a sequence.

CC-2 Coaching versus directing

developing judgment in others

1. Definition and why it matters

Coaching develops a person's judgment, and directing gives them an answer. Each has its place. Managers lean too heavily on directing, and on its most tempting form, rescuing: stepping in with the solution as soon as someone struggles. Rescuing feels helpful and is quicker today. It also teaches people that they are not trusted to think, limits the organisation's capability to the manager's own, and turns people into dependents. The competency consists of asking before telling by default, accepting a slower or different route when it serves someone's growth, and keeping directing for real emergencies, true novices and stakes that leave no room. When the manager does direct, they name it as the exception instead of sliding into it as a habit. It matters because this is how a team comes to be more capable than its manager, and because every other competency in this standard that involves developing someone (delegation, feedback, career growth, leading through leaders) depends on it. The examinations test it through the engineer who brings every decision to the manager, the manager who says "just tell me what to do", the review culture that defaults to blame whatever the policy says, and the organisation that has learned to wait for the executive's answer.

2. Core principles

1. **Ask before telling.** "What options have you considered?" "What would you do if I weren't here?" Questions develop judgment, and an answer takes its place.
2. **Tolerate the slower path.** Someone who reaches a sound decision by a route the manager would not have chosen has learned something that the manager's answer would have stopped them learning. The discomfort of watching is what growth costs.
3. **Direct deliberately and name it.** In an emergency, with a true novice on a task, or where the stakes leave no room, directing is right. The manager says out loud that it is the exception, so that it does not become the default.
4. **Rescuing is the failure that feels like help.** Stepping in at the first wobble ends the learning and teaches the team that ownership can be taken back at any time. The alternative is support that leaves the decision with the person.
5. **Coach the reasoning, especially when asked for the verdict.** A manager who asks to be told what to do is asking to be made dependent. Work through the trade-offs together and decline the trade.

6. **Model it, or it does not spread.** Leaders build a coaching culture by asking for feedback in public, coaching where others can see and being coachable themselves. A mandate from someone who commands will never build one.

---

3. Models and evidence

The practitioner literature on coaching is large and the experimental evidence in management settings is thin. The unit grades its models as practice and borrows one from the leadership literature.

The GROW model [practice]

John Whitmore, Coaching for Performance: GROWing People, Performance and Purpose (1992) sets out a coaching structure of Goal, Reality, Options and Will: what the person wants, where they are now, what they could do and what they will do. For managers its value is that it gives "ask before telling" a shape, with four questions a manager can keep hold of in a one-on-one without training. Its limit is that it structures a conversation. It does not replace the manager's judgment about when a conversation is the wrong tool.

Situational leadership [practice]

This is the model from Paul Hersey and Kenneth H. Blanchard, Management of Organizational Behavior: Utilizing Human Resources (1969), introduced in PL-1 Delegation, ownership, and decision rights and applied here to the choice between coaching and directing. Someone new to a task needs directing, then coaching as their competence grows, then supporting, then delegating. It explains why "always coach" is as wrong as "always direct". A true novice on a task needs direction, and coaching them amounts to abandonment with better manners. Style follows the task, whatever the person's seniority.

The rescue reflex [practice]

This is the failure the unit treats as central, though it is not a named model. The manager instinctively steps in with the solution as soon as someone struggles. It feels like help and it is faster today, and it teaches the person that they are not trusted to think. In an organisation it shows up as the manager of managers doing a manager's job, and at executive level as an organisation that has learned to wait for the executive's answer. Its provenance is experience, and the examinations use it because it is so easy to recognise.

The learning organisation [practice]

Peter M. Senge, The Fifth Discipline: The Art and Practice of the Learning Organization (1990) gave influential form to the idea that an organisation can be designed to learn, turning experience into shared capability instead of leaving it as local memory. In this unit it supplies the practice at executive scope: lessons from incidents that travel between teams, postmortem actions audited until they are done, slack funded for learning, and leadership reviews that take what was learned as seriously as what was shipped. It is graded as practice, and the book's systems-thinking claims are broader than what is used here.

Coaching the coaches [practice]

Again this is not a named model. It is the organisational-scope practice from PL-6 Leading through leaders, seen from the coaching side. Managers develop judgment when someone coaches their reasoning on real situations, which is what the judgment debrief does. The manager of managers should watch for the avoidance patterns of the managers they lead. A manager who cannot deliver a hard truth, or who asks to be told what to do, has a capability gap that will show up everywhere they go.

---

4. Practice

The question first

Whenever someone brings a problem, the manager responds first with a question: what have you considered, what would you do, what is stopping you? If an answer comes at all, it comes after the person has run out of their own thinking. This habit is what makes coaching the default and directing the exception.

The named exception

When the manager directs, they say that they are doing so: "I am going to make this call, because it is an emergency", or "because this is your first one". Naming it stops directing from becoming the default, and it tells the person that ownership will come back to them.

The wobble protocol

When delegated work is struggling, the manager lays out the options before acting. They can support (provide resources, remove a blocker), coach (ask questions that help the person see what they missed) or rescue (take the work over). Rescue is chosen only when being wrong would be costly and irreversible, and then it is done openly and with the reason given, as in PL-1 Delegation, ownership, and decision rights.

The judgment debrief

With a manager who reports to you, the regular one-on-one centres on a hard call they are facing or have just made: what they saw, what they weighed and what they chose. The manager of managers asks and holds back their own answer until the reasoning has run its course. When the manager says "just tell me what to do", they go through the trade-offs together instead.

The peer case discussion

Managers meet as peers to talk through real situations that one of them is facing, and coach each other with questions. It is the cheapest way to build a coaching culture. It only works if the manager of managers models it by asking for feedback in the room, being visibly coachable, and coaching where they might otherwise rule.

Learning loops, audited

At organisational scope the learning system itself gets checked. Do lessons from incidents cross team boundaries? Are postmortem actions closed? Is there slack to learn? Does the leadership review ask what was learned? A learning culture that exists only in policy dies at the first executive review with a flavour of blame, and one visible exception undoes a hundred training sessions.

5. Scaling note

At team scope the object is an engineer's judgment, developed in place of supplying answers. The manager coaches by default, accepts the slower path and directs deliberately. At organisational scope the object is the judgment of managers. The manager of managers coaches the coaches through judgment debriefs, treats a manager's avoidance patterns as the thing to develop, declines to hand down verdicts to a manager who asks for them, and builds the peer practice that turns a coaching culture into something real. At executive scope the object is the learning organisation. Reviews across the whole organisation learn instead of blaming, learning loops are built into how the organisation works and audited, and the coaching cascade runs through every level of leadership. The executive also recognises that an organisation which waits for their answer was taught to wait. The pattern is set out in How Judgment Scales; the managers being coached are the subject of PL-6 Leading through leaders.

6. Judgment

CULTURE & COACHING

- Defaults to coaching in developmental moments, with questions before answers. [team]
- Lets a person take a slower or different path and learn from it. [team]
- Directs only when the stakes or someone's inexperience require it, and names it as the exception. [team]
- Recognises the rescue reflex in themselves and chooses support or coaching instead.
- Coaches a strong engineer who does not mentor, document or lift others, and does not accept that technical strength exempts them. [team]
- Failure mode: rescuing at the first wobble. [team]
- Failure mode: being the source of every answer. [team]
- Failure mode: coaching through a real emergency that needed a call. [team]
- Failure mode: abandoning a true novice in the name of empowerment.
- Coaches a manager who gives only positive feedback on what their kindness is costing, with low-stakes practice and explicit backing through the discomfort. [org]
- Treats a manager's repeated avoidance of a hard conversation as the subject of the coaching, because it will recur everywhere. [org]
- Declines the trade when a manager says "just tell me what to do", and works through the trade-offs with them in place of issuing a verdict. [org]
- Coaches a manager who is too hands-off on the difference between empowering and abandoning: delegation with context, checkpoints and presence. [org]
- Builds the coaching culture through peer case discussions, managers observing and debriefing each other, and visible modelling from the head of the department. [org]
- Failure mode: handing answers to managers and calling it mentoring. [org]
- Failure mode: letting kindness that avoids conflict pass for people skills. [org]
- Failure mode: mandating a coaching culture while modelling command and control. [org]
- Changes reviews that default to blame across the organisation using every tool available: retraining how reviews run, leaders who understand safety in the rooms that matter, public reward for the honest review, and no visible exception left standing. [exec]
- Builds the learning loops into the organisation: lessons that travel between teams, actions audited until they are done, slack to learn, and "what did we learn?" treated as a first-class executive question. [exec]
- Runs the coaching cascade through every level of leadership, so that the organisation's culture teaches itself. [exec]

- Recognises that an organisation which brings decisions fully formed and waits for approval was taught to do so, and changes what the executive does. [exec]
- Failure mode: a learning culture that exists in policy and dies at the first executive review with a flavour of blame. [exec]
- Failure mode: lessons that never cross team boundaries. [exec]
- Failure mode: treating slack for learning as waste to be optimised away. [exec]

7. Tensions

Speed versus development. The answer is quicker today, and the question builds someone who will not need to ask tomorrow. The tension comes up in every one-on-one, and a manager who always settles it in favour of speed ends up with a team that cannot decide.

Coaching versus direction. Coaching all the time abandons the novice, and directing all the time creates dependents. The resolution is situational judgment about the task, the person's experience of it and the stakes. The failure is choosing one mode by temperament.

Support versus rescue. From outside both look like help, and only support leaves the decision with the person. The dividing line is who decides at the end of the conversation, and the manager has to notice which side of it they are standing on.

Kindness versus growth. A manager who spares people any struggle is liked, and the people do not grow. A manager who lets them struggle for what it teaches can seem indifferent. The answer is to stay present through the struggle without ending it, and to explain why.

Policy versus exception. A learning culture can be written into policy, trained and measured, and a single visible review by a senior leader with a flavour of blame wipes it out. The exception sets the culture, not the rule, so the executive's job is to make sure the exception never happens.

8. Worked scenario

A team manager has a capable engineer, eighteen months into the job, who brings every decision to them: which library to use, whether to split a pull request, how to answer a stakeholder's question, whether a design is ready for review. Each question takes the manager thirty seconds, the answers are good, and the engineer carries them out well. The manager has noticed that they are answering a dozen of these a day, and that the engineer's design documents no longer contain recommendations.

The easy path is to keep answering. The questions are quick, the answers are right, and it feels like being a helpful manager. This is the rescue reflex in its mildest and most corrosive form. The manager has become the engineer's judgment, and the engineer has learned that asking is the safe move.

The manager starts by changing their own behaviour. When the next question comes, they ask what the engineer would do and why, and then they wait. The first answers are tentative and hedged. The manager asks what would happen if they were on holiday, and the engineer comes up with a decision and a reason. The manager says "do that". In the weeks that follow they keep to the pattern, answering questions with questions. When the engineer's reasoning is sound the manager says so and nothing more. When it is not, the manager asks what the engineer has not considered, without supplying it. Design documents are now expected to end with a recommendation, and when the manager reviews them they ask about the reasoning behind the recommendation and leave the recommendation itself to the engineer.

The manager does two things deliberately. First, they tell the engineer what they are doing and why ("I have been answering questions you can answer, and I am going to stop, because I want your judgment, not mine, in this work"), so that the change is not taken as withdrawal. Second, they name the exceptions. Anything touching the payment path or a customer commitment should still come to the manager, and the manager will say when a decision falls into that category.

Three months on, the questions are down to two or three a week, and most are ones the manager wants to hear. The engineer has led a design the manager would not have chosen, and it worked. What the manager stopped doing was being helpful, because that helpfulness had been capping the engineer at the manager's own judgment.

9. Related competencies

- PL-1 Delegation, ownership, and decision rights: delegation, where the choice between coaching and rescue is made, and the discipline of not taking work back.
- PL-6 Leading through leaders: coaching the coaches, and the judgment debrief with a manager.
- PL-5 Career development and growth paths: coaching as the form growth takes when the aim is judgment.
- CC-1 Psychological safety and productive conflict: the safety that lets someone answer "what would you do?" honestly.
- DE-3 Reliability, incident response, and operational ownership: the blameless review as the central ritual of a learning organisation.

10. Self-check

1. What does habitual rescuing teach a team? *Answer:* That they are not trusted to think, that ownership can be taken back, and that asking is the safe move. It limits the team's capability to the manager's and creates dependents.
2. When is directing the right choice, and what is the discipline that keeps it from becoming the default? *Answer:* In real emergencies, with true novices on a task, and where the stakes leave no room. The discipline is to name it out loud as the exception ("I'm making this call because...") so that ownership is seen to return afterwards.
3. What are the four questions of the GROW model, and what are they for? *Answer:* Goal, Reality, Options and Will: what the person wants, where they are now, what they could do and what they will do. They give "ask before telling" a shape that a manager can hold on to in a one-on-one.
4. A manager who reports to you says "just tell me what to do". Why is answering them the worst available move? [org] *Answer:* Handing verdicts to a manager makes them permanently dependent, and their team ends up run on your judgment at one remove. Work through the trade-offs together and coach the reasoning; decline the trade.
5. A manager gives only positive feedback to avoid conflict and their team's growth is stalling. What is the coaching? [org] *Answer:* Coach them on what their kindness is costing, meaning the link between honest feedback and their team's development. They practise in low-stakes repetitions, with your explicit backing through the discomfort. The avoidance is itself the development need and should not be treated as a side issue.
6. What does it take to change blame-defaulting reviews across an organisation, and why does one visible exception undo it? [exec] *Answer:* Retraining how reviews are run, putting leaders who understand safety in the rooms that matter, publicly rewarding the honest review that brought the ugly truth out, and leaving no visible exception standing. The exception sets the culture, and one senior review with a flavour of blame teaches the organisation what the real rule is.
7. What does an organisation that brings decisions fully formed and waits for approval tell the executive? [exec] *Answer:* That it was taught to. The fix is for the executive to change what they do, asking questions in place of handing down rulings and pushing decisions down explicitly. Asking people to be more decisive will not fix it.

Sources

-

John Whitmore, Coaching for Performance: GROWing People, Performance and Purpose (1992): the GROW model.

- Paul Hersey and Kenneth H. Blanchard, Management of Organizational Behavior: Utilizing Human Resources (1969): situational leadership; directing as what the novice needs.
- Peter M. Senge, The Fifth Discipline: The Art and Practice of the Learning Organization (1990): the learning organisation.

CC-3 Inclusive practices and equitable opportunity distribution

who gets the stretch work, the visibility, and the benefit of the doubt

1. Definition and why it matters

Managers control how two scarce resources that shape careers are shared out. One is opportunity, meaning the visible work that makes people grow. The other is attention: mentorship, airtime, credit and the benefit of the doubt. If nobody takes care over the distribution, both drift by default to the people who are most similar, most confident, most vocal and nearest, which entrenches inequity and wastes talent. The competency is distributing them deliberately and fairly. Quieter people are heard and credited, the stretch work and the office housework are both spread on purpose, and recognition is traced to contribution instead of visibility. At larger scope it also means designing the structures (location, team membership, calibration, promotion) so that where someone happens to sit does not decide whether they succeed. This is a matter of performance, and charity has nothing to do with it. An organisation that develops all of its talent is stronger than one that develops a favoured subset. The default gradient also always favours the centre, so equity that nobody engineers wears away. The examinations test the competency through the same favourites getting the best projects, the remote team that has become second-class, promotions that follow closeness to the office, and the merger in which the larger culture wins by attrition.

2. Core principles

1. **Opportunity and attention are distributed, whether or not anyone decides.** The only choice a manager has is whether the distribution is deliberate. Left to themselves, both go to the similar, the confident, the vocal and the near.
2. **Track who gets what.** Stretch assignments, visible work, glue work, on-call, airtime and credit form a pattern that only a manager who looks will see, and it is usually different from the one they would have chosen.
3. **Credit follows contribution, not visibility.** The person who presented may not be the person who did the work. The manager's job is to know which is which and to say so.
4. **Location-neutral by design, not by reminder.** A remote team that has become second-class is fixed through structure: redesigned decision-making and information flows, opportunity tracked across sites, and leaders assessed on how their remote members experience the organisation. Asking headquarters to be nicer does not fix it.
5.

Equity is a property of systems at scope. Baselines for promotion, calibration, pay and onboarding have to hold across very different sub-organisations. The populations that the defaults quietly disadvantage, such as acquired teams, remote regions and functions outside headquarters, have to be identified and corrected for.

6. **The composition of the rooms where decisions happen matters.** Whether a leadership team broadens the organisation's thinking is first of all a question of decision quality. An organisation that is diverse everywhere except in those rooms has not solved the problem.

---

3. Models and evidence

The unit draws on one solid research finding, one claim that is honestly contested, and practitioner models whose provenance is clear.

Opportunity and attention as scarce resources [practice]

This is the framing the unit rests on, and it is not a named model. It identifies the two resources a manager hands out that shape careers, and the default gradient along which they flow. It comes from ordinary observation. Its strength is that it turns "be inclusive" into a question a manager can audit and act on: who got the stretch work this quarter, who spoke, who was credited?

Glue work [practice]

Glue work is the essential and under-recognised work that keeps teams functioning, such as coordination, documentation, onboarding and unblocking others. Tanya Reilly, The Staff Engineer's Path: A Guide for Individual Contributors Navigating Growth and Change (2022) names it as work that tends to fall to people who are then not promoted for it. Next to it sits office housework: notes, scheduling, cover. Unless someone manages them, both split along predictable and inequitable lines. The fix is to value the work and share it out, and eliminating it is no answer, because a team with no glue does not become efficient. It gets stuck.

Proximity and promotion [research]

In the randomised experiment reported in Nicholas Bloom, James Liang, John Roberts and Zhichun Jenny Ying, Does Working from Home Work? Evidence from a Chinese Experiment (2015), employees assigned to work from home were more productive and more satisfied, and were promoted less often than colleagues in the office doing the same job. That is direct evidence for the mechanism this unit calls the default gradient: people out of sight fall out of the flow of opportunity, even when they perform better. A later randomised trial

of hybrid work, discussed in the Distributed and Hybrid Work chapter, found no promotion penalty under a hybrid arrangement. That result is the evidence that the gradient comes from design and is not a fixed law.

Diversity and decision quality [contested]

Two claims fall under this heading. Scott E. Page, The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies (2007) argues formally that, on complex problems, groups with diverse perspectives and heuristics outperform groups of uniformly high-ability individuals. Widely cited consultancy studies link workforce diversity to financial performance. The formal argument depends on assumptions about the problem and the group, and the consultancy findings have not survived independent re-analysis, as Jeremiah Green and John R. M. Hand, McKinsey's Diversity Matters/Delivers/Wins Results Revisited (2024) reports. The honest position for this standard has three parts. Homogeneous groups demonstrably converge quickly and can converge on the wrong answer. Who sits in the rooms where decisions are made is a legitimate concern for decision quality. The strong quantitative claims about diversity and performance are not established, and the equity case for this competency does not depend on them.

Merger integration with cultural intent [practice]

This is not a named model either. It is the executive-scope practice the unit treats as the defining case for equity, and it avoids two failures. One is forced assimilation, where the acquirer's culture wins by default and the acquired talent leaves. The other is indefinite coexistence, where two organisations share a name. The practice is to understand both cultures honestly, to define the shared identity and the non-negotiables deliberately, and to integrate with respect and explicit intent while keeping the best of each. The structural half is covered in SV-6 Organisational design and leading change.

---

4. Practice

The opportunity ledger

Each quarter the manager writes down the stretch assignments, the visible work, the conference talks and the customer-facing moments, with who got each one. Then they do the same for the glue work, the on-call, the notes and the cover. The pattern usually differs from the one the manager believed in, and they plan the next quarter's distribution to correct it.

Airtime, managed

In meetings the manager notices who speaks and who does not, asks quiet people directly for their view, and credits each contribution to whoever made it, including when someone else later repeated it more loudly. This happens every time and is not a campaign.

Credit traced

Before giving recognition, the manager asks who did the work, which may not be who presented it. Sometimes the answer is a surprise. The recognition goes to the right person, by name and in public.

Location-neutral decisions

On a distributed team, decisions are made in writing or in meetings every location can attend, information is published instead of passed along in corridors, and the manager checks the opportunity ledger site by site. A leader with remote members is assessed partly on how those members experience the organisation. The Distributed and Hybrid Work chapter has the detail.

The onboarding baseline

A department sets a minimum that every new hire receives (a named guide, a first task they can ship, written context) and lets each team add its own flavour. That way the team someone joins does not decide whether they succeed.

The pattern check across teams

At organisational scope the manager watches which teams, as well as which people, get the strategic work, the visibility and the promotions, and corrects the pattern before it sets. At executive scope the same check runs across sub-organisations, sites and acquired populations, using promotion, pay and attrition data.

---

5. Scaling note

At team scope the question is who on the team gets the stretch work, the visibility and the benefit of the doubt. The manager distributes deliberately and checks their own decisions for affinity. At organisational scope inequity tends to run along location and team membership. The remote team that has become second-class is fixed through structure, onboarding becomes a departmental baseline, belonging across a distributed organisation is engineered instead of hoped for, and opportunity is tracked across teams and sites. At executive scope the object is the organisation's opportunity structure. That includes

equity mechanisms that hold across very different sub-organisations, a leadership team whose composition broadens the organisation's thinking, correction for the populations the defaults disadvantage, and a merger integrated with cultural intent. The pattern is set out in How Judgment Scales; the structural half of a merger is covered in SV-6 Organisational design and leading change.

6. Judgment

- Rotates stretch assignments deliberately and compares the pattern with what they believed it was. [team]
- Manages airtime, for example by asking the quiet person directly, instead of only watching it. [team]
- Traces credit to the person who did the work, who may not be the person who presented it. [team]
- Spreads glue work and on-call fairly and values the glue work visibly.
- Checks their own decisions for affinity: who gets the benefit of the doubt, and why.
- Failure mode: the same favourites getting the best projects. [team]
- Failure mode: the loudest people dominating. [team]
- Failure mode: office housework falling along predictable lines. [team]
- Failure mode: mistaking visibility for contribution. [team]
- Fixes a second-class remote team through structure: decisions and information flows that do not depend on location, opportunity tracked across sites, leaders assessed on how their remote members experience the organisation. [org]
- Engineers belonging across a distributed organisation, with rituals that work in every time zone and norms that do not favour one office. [org]
- Sets an onboarding baseline for the department so that luck in which team someone joins does not decide who succeeds. [org]
- Watches which teams get the strategic work, the visibility and the promotions, and corrects the pattern before it sets. [org]
- Reads promotion, opportunity and attrition data across teams and acts on any inequity the numbers show. [org]
- Holds managers to account for equitable practice on their teams without turning it into a compliance exercise. [org]
- Failure mode: remote-friendly in policy, headquarters-first in practice. [org]
- Failure mode: onboarding quality left to team luck. [org]
- Failure mode: strategic work collecting in the teams nearest the leader. [org]

- Failure mode: belonging initiatives that amount to one office's party streamed to everyone else. [org]
- Failure mode: a manager insisting their team's distribution is fair while the data says otherwise, and nobody checking. [org]
- Integrates a merger with deliberate cultural intent: both cultures understood, the shared identity and non-negotiables defined, the best of both kept. [exec]
- Builds equity mechanisms for promotion, calibration and pay that hold across very different sub-organisations. [exec]
- Treats the make-up of the leadership team as a question of decision quality and changes it over time. [exec]
- Identifies the populations that the organisation's defaults quietly disadvantage (acquired teams, remote regions, functions outside headquarters) and corrects for them. [exec]
- Reads the organisation's opportunity structure from its data and changes the structure as well as the outcomes. [exec]
- Corrects a pay inequity across the organisation when it comes to light, even when the correction is expensive. [exec]
- Failure mode: letting the larger culture win a merger by attrition. [exec]
- Failure mode: equity that holds within teams and breaks between organisations. [exec]
- Failure mode: an organisation that is diverse everywhere except the rooms where decisions are made. [exec]
- Failure mode: cloning yourself at the leadership table. [exec]

7. Tensions

Deliberate distribution versus merit. Rotating stretch work on purpose can look like taking it away from whoever would do it best. Both the best person for the work this quarter and the person who most needs the growth are legitimate criteria. The default gradient is not merit. And an organisation that only ever gives the stretch to people already proven has no means of proving anyone else.

Visibility versus contribution. People who present, speak up and sit near the manager are visible, and visibility is real evidence of something. The judgment lies in working out what it is evidence of, and in not letting it substitute for the work.

Structure versus attitude. Asking the central office to be nicer to remote colleagues costs little and changes nothing. Redesigning decision flows so that location stops mattering is expensive and it works. The difficulty is that the structural fix costs the centre something real, and the centre is where the decision is made.

Integration versus respect. After a merger, integrating quickly destroys what was bought, and coexisting indefinitely leaves two organisations under one name. The resolution is deliberate intent about the shared identity, and the discipline is being honest about what each culture did well.

Evidence versus advocacy. The case for equity is strong, and some of the quantitative claims made in its support are not. This standard argues from the default gradient, from wasted talent and from decision quality. It does not rely on the figures that failed re-analysis, because an overclaimed case is easier to dismiss.

8. Worked scenario

A senior manager runs five teams. Four of them sit with the senior manager at headquarters, and the fifth is based entirely in a second city four time zones away. Over eighteen months the remote team has become second-class on every measure the senior manager can find. Decisions that affect it are made in headquarters meetings it cannot attend. Its engineers have had no promotions while the other teams have had six. It has not been given a strategic project, and two of its strongest engineers have left. Its manager is competent and has raised the problem twice. Both times the response was a reminder to the other managers to include the team.

A reminder is an attempt to change attitudes, and it has now failed twice. The senior manager treats the problem as structural, which it is. The default gradient favours the centre, and nothing in the way the department is designed works against it.

So they change the design. Decisions that affect more than one team move to a written proposal-and-comment process with a fixed window, which gives a team four time zones away the same access as one down the corridor. The two standing cross-team meetings move to an hour every location can attend, are recorded, and have their decisions published. The opportunity ledger is broken down by site for the first time. The pattern it shows (six promotions against none, and no strategic work) goes to all five managers as data, without accusation. The next strategic project goes to the remote team, with a headquarters team as its consumer and not its supervisor. A check of the onboarding baseline finds that the remote team's new hires waited weeks for access that headquarters hires received on their first day, and fixing that becomes a department rule. The five managers' own assessment gains a criterion: how their remote members experience the organisation, measured by asking them.

The senior manager also has the conversation with the remote team's manager that should have taken place the first time the issue was raised. They apologise for the two reminders, explain what has changed in the structure, and say what the manager will be able to point to if things slip. The two engineers who left do not come back. The next two who might have gone stay, and the remote team is included in the next promotion cycle.

The senior manager does not send a third reminder.

9. Related competencies

- PL-3 Hiring and onboarding: interview panels that resemble the manager, and the onboarding that decides whether a new hire becomes effective.
- PL-5 Career development and growth paths: growth paths distributed fairly, and fair promotion across teams.
- PL-4 Performance management: opportunity as part of compensation, and the equity problem within performance decisions.
- SV-6 Organisational design and leading change: the structural half of a merger, and team boundaries as the design that distributes opportunity.
- CC-4 Motivation, recognition, and sustainable pace: recognition and pace on distributed teams, where effort is hardest to see.

10. Self-check

1. Which two resources shape careers, and where do they flow by default? *Answer:* Opportunity (the visible work that makes people grow) and attention (mentorship, airtime, credit, the benefit of the doubt). By default both go to the people who are most similar, most confident, most vocal and nearest.
2. What is glue work, and what is the fix for its inequitable distribution? *Answer:* The essential, under-recognised work that keeps a team functioning, such as coordination, documentation, onboarding and unblocking. It tends to fall to people who are then not promoted for it. The fix is to value it and share it out deliberately. Eliminating it is not the fix.
3. What did the work-from-home experiment find about promotion, and what does it show? *Answer:* Employees assigned to work from home were more productive and more satisfied, and were promoted less often than colleagues in the office doing the same job. It is direct evidence that the default gradient (out of sight, out of the flow of opportunity) operates even when people perform better.
4.

A remote team has become second-class. Why is the fix structural rather than attitudinal? [org] *Answer:* The default gradient favours the centre, and reminders do nothing to a gradient. Decision-making and information flows are redesigned so that location does not matter, opportunity is tracked across sites, and leaders are assessed on how their remote members experience the organisation.

5. What is the honest position on the claim that diverse teams perform better? *Answer:* It is contested. Homogeneous groups demonstrably converge quickly and can converge on the wrong answer, and who sits in the rooms where decisions are made is a legitimate concern for decision quality. The strong quantitative claims linking diversity to financial performance have not survived independent re-analysis, and the equity case does not depend on them.
6. What does a culturally deliberate merger integration look like, and what do both failure modes cost? [exec] *Answer:* Both cultures are understood honestly, the shared identity and non-negotiables are defined deliberately, and integration proceeds with respect and explicit intent, keeping the best of both. Forced assimilation loses the acquired talent through attrition. Indefinite coexistence leaves two organisations under one name.
7. Where does an organisation's equity most often break, and why is the leadership table the test? [exec] *Answer:* Between sub-organisations, between sites, and for the populations the defaults disadvantage: equity that holds within teams breaks across them. The leadership table is the test because an organisation that is diverse everywhere except in the rooms where decisions are made has done nothing to broaden its thinking.

Sources

- Tanya Reilly, The Staff Engineer's Path: A Guide for Individual Contributors Navigating Growth and Change (2022): glue work.
- Nicholas Bloom, James Liang, John Roberts and Zhichun Jenny Ying, Does Working from Home Work? Evidence from a Chinese Experiment (2015): the work-from-home experiment and its promotion penalty.
- Scott E. Page, The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies (2007): the formal argument linking diversity to problem-solving.
- Jeremiah Green and John R. M. Hand, McKinsey's Diversity Matters/Delivers/Wins Results Revisited (2024): the re-analysis of the consultancy findings on diversity and performance.

CC-4 Motivation, recognition, and sustainable pace

including in distributed and hybrid teams

1. Definition and why it matters

Engineers are motivated largely from within, by autonomy, mastery and purpose. Pay and perks prevent dissatisfaction and seldom produce lasting engagement, whereas meaningful work, growth and ownership do. Recognition follows the same rules as feedback: it should be specific, timely, tied to real impact and careful about who gets the credit. Sustainable pace is a leadership responsibility, not a perk, because long-running overwork lowers quality, burns people out and reduces throughput. v3.0 adds distributed and hybrid teams to the title because all three are hardest there. Effort is least visible, recognition is easiest to miss, the line between work and the rest of life is thinnest, and signs of disengagement show up last. The competency is creating the conditions in which people do their best work, and defending a pace at which they can keep doing it, even against the team's own enthusiasm. It matters for several reasons. Burnout usually comes to light at the exit interview. The high-performing team in permanent crunch is the one most likely to be praised and least likely to last. And an executive who treats burnout across the organisation as a wellness problem has misdiagnosed a problem with the plan. The examinations test it through the high performer who has withdrawn, the team whose long hours nobody complains about, the collapse in engagement after a change of manager, and the organisation running on empty after a hard year.

2. Core principles

1. **Cultivate autonomy, mastery and purpose; do not try to buy engagement.** Fair pay is the floor. Above that floor, people are moved by meaningful work, growth and ownership.
2. **Recognise specifically, promptly, and to the right person.** People forget "great job". Naming the contribution and its effect tells them what to repeat. The recognition goes to whoever did the work, not to whoever was visible.
3. **Sustainable pace is the manager's to defend.** An occasional, time-boxed crunch for a good reason may be acceptable, and chronic crunch is a failure of management. A run of heroics shows the system has failed and is nothing to be proud of.
4. **Read the early signals.** Withdrawal, cynicism, ideas that have stopped coming, the strong engineer who has gone quiet. People seldom leave without warning; the signals

come first, to a manager who is paying attention, and on a distributed team they come later and more faintly.

5. **Diagnose before prescribing.** Disengagement has a dozen causes, and only some of them concern work. The first step is a private, open conversation to find out what changed.
6. **At scope, energy is a property of the system.** Burnout across an organisation has a source the executive controls, such as commitments, staffing or the size of the portfolio. The fix goes to the cause: cut the load or extend the time, and do it visibly. Complacency after long success is the opposite illness and is treated the same way, through what the system demands and rewards.

---

3. Models and evidence

Motivation and burnout have been researched more thoroughly than most areas this standard covers, and the grades reflect that.

Autonomy, mastery, and purpose [research]

The self-determination research summarised in Richard M. Ryan and Edward L. Deci, Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being (2000) identifies three needs. In the original they are autonomy, competence and relatedness, and purpose is the popular gloss. The research finds that intrinsic motivation, which drives quality and persistence on complex work, depends on these needs and is undermined by controlling external rewards. Most managers know the framing from Daniel H. Pink, Drive: The Surprising Truth About What Motivates Us (2009), which popularised the research and is graded as a synthesis of it. In practice this means a morale problem is seldom solved with benefits, and benefits will not save a role designed without autonomy, growth or a visible purpose.

Burnout as a job condition [research]

The research programme summarised in Christina Maslach and Michael P. Leiter, Understanding the Burnout Experience: Recent Research and Its Implications for Psychiatry (2016) describes burnout as exhaustion, cynicism and reduced efficacy. It places the causes in the job (workload, control, reward, community, fairness, values) and not in how resilient the individual is. For managers this leads directly to the executive-scope principle. Burnout is dealt with at its cause, by changing the conditions that produced it, and wellness programmes laid on top of an impossible plan treat the person for a problem the research locates in the job. The paper notes that interventions have mostly targeted individual coping, despite the evidence that the causes are situational.

Sustainable pace [practice]

Kent Beck, Extreme Programming Explained: Embrace Change (1999) names the principle, originally as the "forty-hour week", that a team should work at a pace it can keep up indefinitely. Overtime is borrowed from the following week's productivity, and the debt compounds. It is a widely adopted practitioner principle and is graded as one. The evidence for it is the experience of every team that has tried the alternative for more than a few weeks.

Recognition as feedback [practice]

This unit adopts a rule here and does not claim it as a named model. Recognition follows the rules of PL-2 Feedback and difficult conversations, being specific, timely and anchored in behaviour, with one addition: it is traced to contribution instead of visibility. At organisational scope it also rewards what the organisation wants, including shared and invisible work. If only the people who hit individual goals are recognised, everyone learns to drop the glue.

Distributed and hybrid work [practice]

The Distributed and Hybrid Work chapter covers these conditions, and this unit applies them to its competency. Effort is less visible, so the manager has to go looking for things to recognise. The line between work and the rest of life is thinner, so pace is harder to see and easier to break. Signs of disengagement show up later, so the manager's instruments (the one-on-one, the written record, the direct question) matter more. The chapter also holds the evidence on hybrid arrangements, including their effect on retention.

---

4. Practice

The conditions audit

Once a quarter the manager asks three questions about each person. Do they have real autonomy over how they work? Are they growing at something? Can they say why their work matters? A "no" to any of them points to a problem with how the role is designed, and the manager fixes that before anyone reaches for a benefit.

Recognition, specific and traced

When something goes well, the manager names the contribution, its effect and the person who made it, after checking who that was instead of assuming. They do it promptly and in the way that person values. On a distributed team that means asking what happened instead of waiting to see it.

The pace watch

The manager treats the team's hours and heroics as a signal: a run of late nights, a working weekend, someone covering a gap. When a pattern appears, the manager names it to the team, traces it to its cause (staffing, scope, or a culture the team is proud of) and corrects it while the output still looks good. By the time an overworked team complains, its strongest members are already interviewing.

The early conversation

When an engineer who used to be strong has withdrawn, the manager has a private, open conversation within the week and hears, in the engineer's own words, what changed. Nothing is prescribed until the diagnosis has been heard, because withdrawal has a dozen causes and only some of them are the manager's to fix.

The engagement response

When engagement in a team falls sharply, whether after a change of manager, a reorganisation or a hard quarter, the response is measured in weeks. Quarters are too slow. It consists of skip-levels, listening, an honest diagnosis of the cause, and a visible change. "Give it time to settle" is how one bad transition turns into six resignations.

The causal fix at scope

When burnout spreads across an organisation, the executive looks for the source in the plan (the commitments, the staffing, the size of the portfolio) and changes it visibly, with fewer things, more time or more people. Wellness offerings are not withheld, and they are not presented as the fix.

---

5. Scaling note

At team scope the object is one team's motivation and pace, whether the team shares a room or is distributed. The manager cultivates the intrinsic drivers, recognises specific contributions and defends the pace against the team's own enthusiasm. At organisational scope the cases arrive as signals passed through the layer of managers: the high-performing team quietly burning out, engagement collapsing after a change of manager, the disengaged engineer three teams away, and a recognition system that has to reward shared and invisible work across teams. At executive scope motivation and pace are conditions of the organisation. Burnout is traced to the plan that caused it and the plan is changed. Complacency after long success is treated as a strategic risk. Ambition is funded again, with real safety for smart failure. The organisation's energy is managed through what the system demands and rewards, which works where speeches from the stage do not. The

pattern is set out in How Judgment Scales; the manager's own energy and boundaries are covered in CC-6 Self-management.

---

6. Judgment

- Cultivates autonomy, mastery and purpose deliberately, and fixes role design before reaching for perks. [team]
- Recognises specific contributions promptly, in the way the person values, and credits whoever did the work. [team]
- Defends a sustainable pace, including against the team's own enthusiasm. [team]
- Spots burnout and disengagement early from changes in behaviour, and attends to the person before the output. [team]
- Goes looking for effort on a distributed team instead of waiting to see it, and keeps an eye on the thinner line between work and the rest of life.
- Failure mode: trying to motivate with perks while starving autonomy and purpose. [team]
- Failure mode: generic praise. [team]
- Failure mode: treating permanent crunch as normal. [team]
- Failure mode: finding out about burnout at the exit interview. [team]
- Failure mode: taking a distributed team's silence for contentment.
- Corrects the long-hours culture of a high-performing team while the metrics still look good: names the pattern to the manager, traces the cause and fixes it before people leave. [org]
- Responds within weeks to a sharp fall in engagement after a change of manager, with skip-levels, listening and an honest diagnosis. [org]
- Meets an engineer who is disengaging with questions before judgment. [org]
- Builds a recognition system that rewards team outcomes and invisible contributions such as glue, mentoring and help across teams, as well as people who hit individual goals. [org]
- Coaches a manager whose distributed team feels disconnected towards deliberate norms and rituals, because distributed teams do not hold together by accident. [org]
- Recognises burnout across a department as a property of the system and changes the system. [org]
- Failure mode: riding the burning team because its output is high. [org]
- Failure mode: waiting out a collapse in engagement. [org]
-

Failure mode: managing a disengaged person's symptoms without asking what happened. [org]

- Failure mode: incentives that quietly punish collaboration. [org]
- Traces burnout across the organisation to the plan that caused it and changes the plan visibly, with fewer commitments, more time or more people. [exec]
- Treats long success that has bred complacency and aversion to risk as a strategic risk, and restores energy through structure: a renewed mission, explicit room for ambitious bets, safety for smart failure re-established, and reward for the attempt as well as the win. [exec]
- Shapes how the organisation relates to pace through what leadership rewards, what it survives and what it refuses to treat as normal. [exec]
- Keeps people motivated through a long, hard period by telling the truth about it. [exec]
- Bases a decision on returning to the office or working remotely on the evidence about the work, and not on reasons unconnected to it. [exec]
- Failure mode: wellness theatre laid over an impossible portfolio. [exec]
- Failure mode: mistaking long success for permanent health. [exec]
- Failure mode: demanding innovation while punishing every miss. [exec]
- Failure mode: burning the organisation's best people as though they were a renewable resource. [exec]

7. Tensions

Enthusiasm versus pace. The team wants to push, the launch is close, and a manager who defends the pace against what the team wants looks like an obstacle. The judgment is in telling a time-boxed push for a good reason apart from a culture of crunch, and in stopping the push when the time is up.

Output versus health. The high-performing team with a long-hours culture is delivering, so acting means acting before anything visibly goes wrong. The failure is waiting for someone to complain, which happens only after the strongest people have started interviewing.

Recognition versus fairness. Praising one person in public can demotivate the people whose invisible work made the result possible. Tracing credit and recognising shared work resolve it. What makes it hard is that the visible contribution is the easier one to see.

Autonomy versus alignment. Autonomy is a primary driver of motivation, and an organisation still needs its teams pulling in the same direction. As in DE-5 Designing how work flows, the line is drawn by setting direction and boundaries centrally and leaving methods to teams. Autonomy over how the work is done is what motivates, and it does not require autonomy over what is done.

Honesty versus morale. In a hard period the temptation is to tell people it will be fine. What keeps people motivated is the truth about where the organisation stands, what is being done and what they can expect. Reassurance uses up trust, and honesty gives people something they can keep working for.

8. Worked scenario

A team manager runs a high-performing team that has spent a quarter in crunch. A launch slipped, the team made up the time with evenings and weekends, the launch landed, and leadership praised the team's commitment at an all-hands. Now a second initiative is behind schedule and the team is doing the same again. Nobody has complained. The two strongest engineers are working the latest. The product partner has said, approvingly, that this team "just gets it done".

The temptation is to let it carry on. The output is excellent, the team is proud of it, and stepping in looks like punishing success. That reading is the failure this unit describes. By the time an overworked team complains, its strongest members are already interviewing, and the praise at the all-hands has just told the whole organisation what leadership rewards.

At the retrospective the manager names the pattern to the team as a failure of the system, and tells them it is nothing to wear as a badge. Two quarters in a row rescued by heroics means the plan was wrong twice and the team paid for it. The manager traces the cause. Commitment and laziness have nothing to do with it. The team's capacity has been planned at a hundred percent with no slack, so every surprise has turned into overtime. The manager fixes the plan and leaves the people alone. The second initiative's scope is worked out again from what the team can deliver at a sustainable pace, and the product partner gets the trade-off in DE-1 Prioritisation under constraint's terms: this scope by this date at this pace, or a smaller scope by the same date. The two strongest engineers each get a private conversation about what the pace has cost them, and hear that the plan which demanded it was the manager's responsibility and not theirs.

The manager also does the harder thing, which is to go upward. They tell their own manager that the praise at the all-hands, though well meant, rewarded exactly the pattern that will cost the team its best people, and they ask that the next recognition go to a team that delivered at a pace it could sustain.

The initiative ships smaller and on time, and the team gets its evenings back. Months later, one of the two strongest engineers says they had been about to leave. The manager never accepts "just gets it done" as a compliment.

9. Related competencies

- PL-5 Career development and growth paths: the one-on-one where disengagement first shows, and growth as the deepest driver.
- PL-2 Feedback and difficult conversations: recognition as feedback that is specific, timely and anchored in behaviour.
- CC-6 Self-management: the manager's own pace and boundaries, and the load that degrades their judgment.
- CC-3 Inclusive practices and equitable opportunity distribution: who gets recognised, and the invisible work that holds the team together.
- DE-1 Prioritisation under constraint: the plan at a hundred percent with no slack, and the cost of saying yes.

10. Self-check

1. Why do perks not substitute for autonomy, mastery and purpose? *Answer:* Pay and perks prevent dissatisfaction but do not create engagement. Intrinsic motivation, which drives quality and persistence on complex work, depends on autonomy, growth and a visible purpose, and benefits will not save a role designed without them.
2. What does the burnout research locate as the cause, and what does that imply for the manager? *Answer:* It locates the cause in the job (workload, control, reward, community, fairness, values) and not in the individual's resilience. Burnout is therefore dealt with at its cause, by changing the conditions that produced it. Wellness offered on top of an impossible plan treats the person for a problem in the job.
3. What makes recognition repeatable rather than forgettable? *Answer:* Naming the specific contribution and its effect, promptly, to the person who did it, in the way that person values. "Great job" tells nobody what to do again.
4. Why act on a burning-out team whose output is still excellent and where nobody is complaining? [org] *Answer:* By the time an overworked team complains, its strongest members are already interviewing. The pattern is named to the manager, traced to its cause (staffing, scope, or a culture the team is proud of) and corrected while the metrics still look good.
5. Engagement dropped sharply in one team after a manager change. What is the response, and what is the failure? [org] *Answer:* Speed and curiosity, within weeks:

skip-levels, listening, an honest diagnosis of the new manager's effect, and a visible change. The failure is "give it time to settle", which is how one bad transition turns into six resignations.

6. Burnout is spreading across the organisation during an extended high-pressure period. Why is the executive's responsibility causal rather than palliative? [exec] *Answer:* The pressure has a source the executive controls, in the commitments, the staffing and the size of the portfolio. Cut the load or extend the time, and do it visibly. Wellness programmes cannot make up for an impossible plan, and presenting them as the fix is theatre.
7. An organisation that has succeeded for years has stopped taking smart bets. What is the diagnosis and the structural fix? [exec] *Answer:* The diagnosis is complacency and aversion to risk, which is a strategic risk. The fix is to renew the sense of mission, make explicit room for ambitious bets with safety for smart failure re-established, bring in fresh challenges and people, and reward the attempt as well as the win. All of it works through the system, and speeches will not do it.

Sources

- Richard M. Ryan and Edward L. Deci, Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being (2000): self-determination and intrinsic motivation.
- Daniel H. Pink, Drive: The Surprising Truth About What Motivates Us (2009): autonomy, mastery and purpose as the popular synthesis.
- Kent Beck, Extreme Programming Explained: Embrace Change (1999): sustainable pace.
- Christina Maslach and Michael P. Leiter, Understanding the Burnout Experience: Recent Research and Its Implications for Psychiatry (2016): burnout as a condition of the job.

CC-5 Values, integrity, and ethical judgment under pressure

what you do when the right call is the costly one

1. Definition and why it matters

A culture is the accumulated pattern of which behaviour gets rewarded, tolerated and punished, and a values poster is no part of it. What defines it is what leaders do when the stated values become costly. Everyone watches the gap between word and action, and everyone believes the action. v1.0 framed this competency as modelling values under pressure. v3.0 extends it to ethical judgment, because the situations that test a manager hardest are those where the organisation's stated values, the law, the policy and the pressure of the moment leave the question open and the manager has to settle it. The competency is upholding values at exactly the point where doing so is expensive. That means dealing with the brilliant jerk instead of banking their output, reporting status truthfully without shading it, and obeying the error budget when the launch is due. At scope it also means running culture as the system it is, made of what leadership visibly does, rewards, promotes and tolerates. It matters because tolerating a behaviour endorses it, because a value with an exemption for seniority has become a price, and because one visible exception undoes quarters of work. The examinations test it through the high performer whose behaviour buys silence, the manager asked to report a slipping project as on track, and the leadership that shipped through an exhausted error budget in full view of everyone.

2. Core principles

1. **The costly moment is the culture.** Values that last only while they cost nothing are decoration. When the right call costs a launch, a customer or a person, what the leader does is what the organisation learns and copies.
2. **Tolerating is endorsing.** Behaviour that is rewarded gets repeated, and behaviour that is tolerated gets learned. Banking the brilliant jerk teaches everyone that results buy exemption.
3. **Consequences reach the powerful, or they are pricing.** A value enforced only below a certain level has stopped being a value. A senior, high-impact leader who breaches one faces real and visible consequences, and how results were achieved counts as part of the result.
4.

Do not shade the truth upward, and absorb the cost of declining. When asked to report a status, a rating or a commitment as better than it is, the manager declines. The manager carries the cost of declining and does not pass it down to the team.

5. **Behaviour changes when the system changes.** A third mandate for something two mandates did not fix will fail. Behaviour shifts when the desired thing becomes the easy path, is visibly valued, and takes less effort to do. Culture change means realigning hiring, promotion, recognition and consequences over many quarters, and an announcement will never achieve it.
6. **Where the rules do not settle it, judgment does — and it is explained.** A request may be legal and wrong, a policy may produce a bad outcome in this particular case, or one value may pull against another. The manager decides on the merits, gives the reasons and owns the decision.

3. Models and evidence

The research literature on culture is deep, and the experimental evidence on how to change it is thin. The unit grades its models as practice and takes its working definition from the founding work in the field.

Artifacts, espoused values, and assumptions [practice]

Edgar H. Schein, Organizational Culture and Leadership (1985) describes culture at three levels: the visible artifacts, the values the culture espouses, and the underlying assumptions that drive behaviour. Those assumptions are usually unspoken and often contradict the espoused values. This unit uses the model to diagnose. Where the espoused values and the behaviour diverge, the behaviour is the culture. The assumptions it exposes, such as "results buy exemption" or "the budget is obeyed unless it is inconvenient", are what a leader is in fact managing.

Culture as what is rewarded, tolerated, and punished [practice]

This is less a named model than the working definition this standard uses. Culture is the accumulated pattern of how people behave, and it is set far more by what leaders visibly do, reward, promote and tolerate than by stated values, policies or announcements. For change, it follows that culture is realigned through the systems of hiring, promotion, recognition and consequences. A single visible exception, such as the brilliant jerk who is tolerated or the value waived for a launch, can undo quarters of work.

Service-level objectives and error budgets [practice]

This is the mechanism from Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy (editors), Site Reliability Engineering: How Google Runs Production Systems (2016), introduced in DE-3 Reliability, incident response, and operational ownership and treated here as the unit's defining test case. An error budget turns "reliability first" from a slogan on a poster into a rule agreed in advance, and it is worth only as much as leadership's willingness to obey it when a launch is due. Shipping through an exhausted budget in front of the whole organisation teaches everyone the real policy. Obeying it at a cost is the most legible cultural act an engineering leader can perform.

Leading change [practice]

SV-6 Organisational design and leading change uses the model from John P. Kotter, Leading Change (1996) for structure, and this unit uses it for norms. Credible leaders form a coalition and model the new behaviour, systems are realigned behind it, early wins are made visible, persistence is measured in quarters, and the change is anchored in how decisions are made. In culture change the steps most often skipped are the last ones, consolidation and anchoring. The announcement only starts the process.

The costly moment [practice]

The unit is organised around this concept, which is not a named model. It describes a situation where a stated value and the pressure of the moment point in different directions and people are watching the decision. It comes from experience, and because it is so easy to recognise, nearly every scenario in this competency is built from one.

---

4. Practice

The values audit, in behaviour

Once a year the manager lists the organisation's stated values and writes beside each one what happened the last time it was costly. Where the two diverge, the divergence is the culture, and the manager names it as such before changing anything.

The brilliant-jerk decision

When a high performer's behaviour breaches the team's values, the manager acts within the month. They give specific feedback, set an explicit expectation and a date, and let consequences follow through PL-4 Performance management if the pattern continues. Strong output is no reason to wait, since every week of tolerance announces the real values.

Declining to shade

When asked to report a project, a rating or a forecast as better than it is, the manager says no plainly, offers the honest picture with options, and bears the discomfort personally instead of passing it to the team. This habit is what makes the manager's good news believable.

Changing behaviour by changing the system

For a norm the organisation wants, whether documentation, blameless reviews or a practice spreading from one team to others, the manager makes it the easiest route to something people already want. They value it visibly in recognition and promotion and reduce the friction. Nobody issues a mandate for what two mandates failed to fix.

Consequences at the top

When a senior leader breaches a core value, the consequence is real and visible, and how they achieved their results is treated as part of their performance. The executive begins with the mirror. If the leadership team models behaviour the organisation should not copy, that is the first culture problem, and it is handled with the directness expected of any manager.

Owning the contradiction

Sometimes leadership itself breaches a stated value, by shipping through the budget or waiving the process for a launch, and everyone notices. The only repair is to own the contradiction publicly and behave differently at the next costly moment. Anything less turns the value into décor.

---

5. Scaling note

At team scope the object is the example set in the room: values upheld when they cost something, credit given, mistakes owned, the brilliant jerk dealt with. At organisational scope the manager learns how culture works. Behaviour is the culture, it is set by what managers do every day, it changes when the system changes, and practices spread by pull. The manager also enforces the values between people, for instance by backing a new manager against a senior engineer who undermines them. At executive scope culture is the executive's system. The costly moment is decided in line with the stated value, or the contradiction is owned. Consequences reach the powerful. Culture change runs as a realignment over many quarters. Well-loved but costly norms are evolved together with their champions. Culture is built into the institution so that it survives growth, and direction is carried by line leaders who visibly believe in it. The pattern is set out in How Judgment

Scales; the safety that lets people say when a value has been breached is covered in CC-1 Psychological safety and productive conflict.

6. Judgment

- Upholds values at exactly the moments they are costly, knowing the team is watching. [team]
- Gives credit, owns mistakes and respects boundaries, so that actions match stated principles. [team]
- Deals with a values breach by a top performer instead of banking the output. [team]
- Recognises when they are being asked to shade the truth about status, and declines.
- Failure mode: values that evaporate under deadline pressure. [team]
- Failure mode: saying one thing and rewarding another. [team]
- Failure mode: assuming you only model behaviour when you mean to. [team]
- Reads culture from behaviour instead of posters, and treats any divergence from the stated values as the real culture. [org]
- Changes behaviour by changing incentives and friction, and does not issue a third mandate for what two failed to fix. [org]
- Spreads a good practice by pull (visible value, easy adoption, champions) and avoids decree. [org]
- Deals with a respected senior engineer who quietly undermines a newly promoted manager directly and in private, calling the behaviour what it is, while backing the manager in full view. [org]
- Holds managers to the organisation's values, including the manager who keeps a toxic high performer because the numbers are good. [org]
- Declines to shade status, ratings or commitments upward, and absorbs the cost of declining instead of passing it down. [org]
- Failure mode: a third mandate for something two mandates did not fix. [org]
- Failure mode: expecting a values statement to outvote the reward system. [org]
- Failure mode: leaving a new manager to survive a senior saboteur alone. [org]
- Decides the costly moment in line with the stated value, for example by obeying the error budget with the launch due, or owns the contradiction publicly and behaves differently next time. [exec]
- Makes sure consequences reach the powerful: a senior leader who breaches a core value faces real and visible ones, and a director who hits every number through fear is acted on for how they did it. [exec]

- Starts with the mirror and fixes the behaviour the leadership team itself models first, because the organisation copies the top table. [exec]
- Runs culture change as a realignment of the system over many quarters, with credible leaders modelling it, incentives and promotions realigned, early wins made visible and the effort sustained, and treats the announcement as no more than the starting gun. [exec]
- Evolves a well-loved but costly norm without breaking trust, by honouring what it did well, involving its champions in what replaces it and being honest about what changed and why. [exec]
- Builds culture into the institution so that it survives growth beyond a few hundred people, with leaders hired and promoted for it, norms taught and systems aligned. [exec]
- Carries direction through line leaders who visibly believe it; if direction dies two levels down, it was announced and never communicated. [exec]
- Exercises ethical judgment where the law, the policy and the pressure leave the question open, and explains the decision. [exec]
- Failure mode: shipping through the error budget and expecting the poster to survive. [exec]
- Failure mode: values with an exemption for seniority. [exec]
- Failure mode: culture change by announcement at an all-hands. [exec]
- Failure mode: scaling by dilution. [exec]
- Failure mode: mistaking cascade emails for communication. [exec]

7. Tensions

Results versus methods. An individual or a leader who delivers while breaching the values forces the question of what performance means. This standard's answer, as in PL-4 Performance management, is that the methods are part of the result. The pull towards the number is strong, and the examinations test whether the candidate resists it.

Consistency versus context. Values applied identically every time can produce a bad outcome in a case the rule did not anticipate, and values applied with judgment can look like values applied selectively. The resolution is to decide on the merits, give the reasons and own the decision. The explanation is what separates judgment from exemption.

Change versus trust. Changing an entrenched norm, even a costly one, breaks something people were attached to. Evolving it with its champions, honouring what it did well and being honest about the reasons takes longer than announcing a new norm, and it is the only route that does not use up trust.

Honesty upward versus safety downward. Refusing to shade a report costs the manager standing with their own leadership. Pushing the team to make the report come true costs the team. The manager takes the cost, and taking it is the competency.

Rules versus judgment. Some requests are legal and wrong, and some policies would cause harm in a particular case. The rule leaves these open and the manager's judgment has to close them. Judgment is harder to defend than compliance, so the manager explains the reasoning as carefully as the decision itself.

8. Worked scenario

A head of engineering leads an organisation whose stated value, painted on the wall and repeated at every all-hands, is "reliability first". Six weeks ago a major launch fell due while the core service's error budget was exhausted. The organisation's agreed rule (budget exhausted, so stabilise) meant the launch should have waited. The chief executive wanted the date and the head of engineering agreed to ship. The launch went out, and a week later the core service had its worst incident of the year. Everyone in the organisation saw the sequence. Since then two teams have quietly stopped tracking their error budgets, and a principal engineer has said in a skip-level that "reliability first" has become a joke.

The cultural significance is total. What the organisation took away was the real value, that the budget is obeyed unless a launch is due, and the importance of the launch did not register. Two easy responses are available. One is to say nothing and let the incident fade. The other is to restate the value with a new policy and firmer wording. The first confirms the lesson. The second is a third mandate for something the organisation has just watched leadership break, and it will be read in exactly that light.

The only repair is to own the contradiction in public and act differently at the next costly moment, and the head of engineering does both. At the next all-hands they describe plainly what happened. The organisation had a rule, leadership broke it for a launch, the incident followed, and the two teams that stopped tracking their budgets drew the right conclusion from what they saw. They say the contradiction was their own, since the head of engineering agreed to ship, and that the value is worthless unless it holds when it is expensive. They do not explain why the launch mattered, because the organisation already understood that and it is beside the point.

Then comes the behaviour. The head of engineering reinstates the budget rule with one change. A decision to ship through an exhausted budget can only be taken at the executive table, in writing, with the reason and the accepted risk recorded and published to the organisation. The rule is tested four weeks later, when a second launch meets a red budget on a different service. The head of engineering holds the launch, at a visible cost to a customer commitment, and says so. The chief executive hears the reasons in advance,

set out in terms of both the organisation and the business: the value exists only if it costs something, and the last time it was waived the organisation paid more.

The two teams start tracking their budgets again without being asked. The principal engineer says nothing, which is a signal in its own right. The head of engineering does not send round a reminder about the value. It would have been the cheapest response, and it would have finished the value off.

9. Related competencies

- PL-4 Performance management: the brilliant jerk, and the leader who delivers through fear, treated as performance problems.
- CC-1 Psychological safety and productive conflict: the safety that lets people say a value has been breached, and trust as a behavioural account.
- DE-3 Reliability, incident response, and operational ownership: the error budget as the mechanism, and the review after the failure.
- DE-2 Estimation, planning, and honest commitments: declining to give a false forecast under pressure.
- SV-6 Organisational design and leading change: leading change through structure, and evolving norms with their champions.

10. Self-check

1. Why do teams believe actions over stated values? *Answer:* Culture is the accumulated pattern of what is rewarded, tolerated and punished in practice, and the costly moment exposes it. A value that only holds while it costs nothing tells people nothing. What a leader does when it is expensive is what they learn and copy.
2. What does tolerating a brilliant jerk teach a team? *Answer:* That results buy exemption from the values. Each week of tolerance announces the real culture more loudly than any statement. The fix is feedback, an expectation, a date and consequences, all within the month.
3. You are asked to report a slipping project as on track. What do you do, and who carries the cost? *Answer:* Decline plainly, and give the honest picture with options. The manager bears the cost of declining and does not push the team to make the report come true.
4. Two documentation mandates failed. What does the third attempt have to change, and why? [org] *Answer:* It has to change the system, and a new mandate will not do that. Documentation becomes the easiest route to something people already want

(onboarding, review, incident response), writing it is visibly valued, and keeping it up takes less effort. Behaviour changes when the system changes, and a third mandate for what two did not fix will fail the same way.

5. A respected senior engineer is subtly undermining a newly promoted manager. What is the response? [org] *Answer:* Speak to the engineer directly and in private, calling the behaviour what it is, while backing the manager in full view. Letting it pass teaches the department that anyone with tenure can negotiate with authority.
6. Leadership shipped through the error budget and everyone noticed. What did the organisation just learn, and what is the only repair? [exec] *Answer:* It learned the real value: the budget is obeyed unless a launch is due. The only repair is to own the contradiction publicly and behave differently at the next costly moment. A reminder or a new policy turns the value into décor.
7. A director hits every number and leaves a trail of burned-out people and fear. Why must the executive act, and on what? [exec] *Answer:* The executive acts on how the results were achieved. Every quarter of tolerance announces the organisation's real values, and a value enforced only below a certain level is a price and has stopped being a value. Consequences have to reach the powerful.
8. How does culture survive growth past a few hundred people? [exec] *Answer:* Through deliberate institutionalisation, with leaders hired and promoted for it, norms taught and systems aligned. At that size the founders' pull no longer reaches everyone, and dilution is a responsibility for leadership to manage, not weather to be endured.

Sources

- Edgar H. Schein, Organizational Culture and Leadership (1985): artifacts, espoused values and underlying assumptions.
- Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy (editors), Site Reliability Engineering: How Google Runs Production Systems (2016): error budgets as a rule agreed in advance, and what obeying them costs.
- John P. Kotter, Leading Change (1996): the steps of change, and the ones most often skipped.

CC-6 Self-management

the manager's own time, energy, boundaries, and resilience

1. Definition and why it matters

Self-management is the manager turning their judgment on themselves: how they spend their time, where their energy goes, which boundaries they hold, and how they keep thinking clearly under load. No v1.0 competency took the manager as its object, and the previous edition of this standard treated an overloaded manager as something for their own leader to notice. v3.0 makes it a competency because a manager who cannot manage their own time, energy and boundaries cannot keep up any of the other twenty-eight. One-on-ones are the first thing to be cancelled. The difficult conversation is put off because there is no energy for it. Decisions get made to end a meeting instead of to get the answer right. And the team reads the manager's state and copies it. It matters because the failure cannot be seen from inside. An exhausted manager does not feel their judgment slipping; they feel that their days are full. At scope, a leader's state also becomes the organisation's state. The examinations test it through the new manager working evenings to keep up with two jobs, the calendar full of meetings with no one-on-ones in it, the senior manager who realises they have snapped at two people who had done nothing to deserve it, and the executive who sees that the organisation has grown anxious and traces the anxiety back to themselves.

2. Core principles

1. **Your time is the team's scarcest resource, and the inbox does not get to allocate it.** Each week the manager decides where the hours go, measured against what the team needs, instead of finding out on Friday where they went.
2. **Leverage decides what you do yourself.** An hour on something only you can do, such as the delegation brief, the hard conversation or the decision, is worth ten hours on things others could do. Left alone, a calendar fills up with the second kind.
3. **Boundaries are a management practice, not a personal indulgence.** The hours you keep, the recovery you take and the load you turn down are what keep your judgment usable. They also show the team what is permitted.
4. **Your state is a signal the team reads.** People copy anxiety, exhaustion and irritation before anyone notices them. A manager who snaps at two people who did nothing to deserve it has one problem, which is load, and it is a mistake to treat it as two problems with people.
5.

Recognise your own overload early and ask for help before it reaches the team. A manager who soaks up everything until they break has handed the cost to the very people they meant to protect.

6. **Separate your anxiety about a situation from the decision it needs.** The urge to act, to answer or to make the discomfort stop tells you nothing about what the situation requires. The decision is made on the situation's terms.

---

3. Models and evidence

The unit draws on research into work and recovery, on one body of findings that is honestly contested, and on practitioner writing about the manager's own effectiveness.

Managing your own time [practice]

Peter F. Drucker, The Effective Executive (1967) sets out the discipline. Find out where your time goes by recording it instead of guessing. Cut whatever does not need doing at all. Gather what remains into blocks long enough to use, and put first things first, which means deciding what not to do. Drucker wrote for executives, and the advice holds for any manager. His central finding is that people are reliably wrong about where their time goes until they measure it, and that is why the first practice in this unit is a time log.

Managerial leverage [practice]

This is Grove's framing, introduced in PL-1 Delegation, ownership, and decision rights. A manager's output is the output of their organisation, and some activities yield far more than others. In self-management it becomes the test for what the manager keeps doing. The high-yield activities, such as a clear delegation, a hard conversation or a decision only they can make, are protected. The low-yield activities that fill a calendar by default are delegated, declined or batched.

Burnout as a job condition [research]

The research summarised in Christina Maslach and Michael P. Leiter, Understanding the Burnout Experience: Recent Research and Its Implications for Psychiatry (2016) applies to managers too. Burnout shows as exhaustion, cynicism and reduced efficacy, and it is caused by the conditions of the job (workload, control, reward, fairness), not by a shortage of resilience. So a manager's own burnout is dealt with by changing their load and their control over it. A manager who treats their exhaustion as a character flaw to overcome has misread the research in the same way as an executive who offers wellness on top of an impossible plan.

Recovery and detachment [research]

Sabine Sonnentag and Charlotte Fritz, The Recovery Experience Questionnaire: Development and Validation of a Measure for Assessing Recuperation and Unwinding from Work (2007) distinguishes four experiences through which people recover from work: psychological detachment from work outside working time, relaxation, mastery experiences, and control over one's own time. Of the four, detachment showed the strongest relation with well-being. The study is cross-sectional, so it shows association rather than cause. For managers this means a boundary consists in detaching from work, and stopping work is not enough on its own. A manager who answers messages every evening has rested, perhaps, but has not recovered.

Ego depletion and decision fatigue [contested]

Roy F. Baumeister, Ellen Bratslavsky, Mark Muraven and Dianne M. Tice, Ego Depletion: Is the Active Self a Limited Resource? (1998) and the work that followed it claim that self-control and decision quality draw on a limited resource that runs down over the day, so later decisions are worse. The large multi-laboratory replication in Martin S. Hagger, Nikos L. D. Chatzisarantis and 47 co-authors, A Multilab Preregistered Replication of the Ego-Depletion Effect (2016) did not find the effect. The honest position is that the strong claim lacks support, and managers should not organise their lives around a resource that may not exist. The practical advice that survives (make important decisions when rested, and do not decide something just to end a discomfort) rests on the recovery research and on ordinary experience, and does not need depletion to be true.

The manager’s own resilience [practice]

Lara Hogan, Resilient Management (2019) gives a practical treatment of the manager's own state as their team goes through change: noticing one's own reaction, building a support network made up of people other than one's reports, and asking for help. Here it supports the executive-scope practice of building the support that sustained judgment depends on. It is also a reminder that this support belongs with a leader's peers. The team is the wrong place for it.

---

4. Practice

The time log

For two weeks the manager records every hour as it is spent. They compare the log with what the team needs (one-on-ones, the hard conversations, the decisions only the manager can make), and the gap is the first thing to change. The log is repeated once a year, because calendars drift.

The leverage sort

Every recurring activity goes into one of three groups: only I can do this, someone else could do this, or this does not need doing. The first group is protected in the calendar, the second is delegated with a brief, and the third is stopped. When a manager's calendar is all meetings, they have usually skipped the sort.

The protected hours

One-on-ones, and time to prepare for hard conversations, are fixed in the calendar and are not moved for status. Two hours a week stay unscheduled for whatever will come up. The team can see the protection, and that is part of the point.

The boundary, kept and shown

The manager decides which hours they work and which they do not, sticks to them, and lets that be seen. Messages go out in the morning instead of at midnight, and holidays are taken without interruption. What the manager does is what the team learns is allowed.

The state check

Once a week the manager asks privately: have I been short with anyone who did not deserve it, have I decided anything just to end a discomfort, have I cancelled a one-on-one? A yes to any of these counts as a signal of overload and is acted on that same week, by delegating something, declining something or asking for help.

The support network

The manager builds relationships with peers, and with a mentor outside their reporting line, and uses them for the things that cannot go to the team or upward. At executive scope this has to be deliberate, because the executive's peers are the only people who can hear these things.

---

5. Scaling note

At team scope the object is the first-line manager's own time, energy and boundaries: the calendar, the evenings, and the one-on-ones that get cancelled first. The manager

sets them on purpose and is seen to keep them. At organisational scope the object is the manager's effectiveness as the layer between teams and executives. The pull is to be in every room, load starts to degrade judgment about people, and short temper lands on managers who did nothing to earn it. The practice is to change the load before it changes the decisions. At executive scope the object is the leader's own resilience when consequence and visibility reach the whole organisation, along with the support that makes sustained judgment possible. The leader also has to recognise that their state becomes the organisation's. An anxious executive produces an anxious organisation, and the repair begins with the executive. The pattern is set out in How Judgment Scales; the team's pace, for which the manager's own pace sets the example, is covered in CC-4 Motivation, recognition, and sustainable pace.

---

6. Judgment

- Manages their own time and calendar so that the hours go where the team needs them and the inbox does not decide. [team]
- Protects one-on-ones and preparation for hard conversations against status pressure. [team]
- Spots their own overload and burnout early, sets boundaries and asks for help before the team feels it. [team]
- Keeps working hours deliberately and visibly, so the team learns what is allowed. [team]
- Separates their own anxiety about a situation from the decision it needs.
- Failure mode: the new manager doing both jobs by working evenings, and visibly exhausted. [team]
- Failure mode: a calendar full of meetings with no one-on-ones in it. [team]
- Failure mode: treating exhaustion as a character flaw to overcome. [team]
- Failure mode: deciding in order to end the discomfort instead of to get it right.
- Manages their own effectiveness as the layer between teams and executives with deliberate care for time, attention and the temptation to be in every room. [org]
- Recognises when overload is degrading their judgment about people, and changes the load before it changes the decisions. [org]
- Notices that they have been short with managers who did not deserve it and reads that as a signal of overload, not as two problems with people. [org]
- Helps an overloaded manager who reports to them shed load before it breaks them, and does the same for themselves. [org]
-

Failure mode: forty hours of meetings a week and no proper one-on-one with a manager in a month. [org]

- Failure mode: absorbing every escalation until the manager's job has quietly become theirs, and calling it support. [org]
- Failure mode: a support network made up of one's own reports. [org]
- Keeps their own judgment and resilience intact under consequence and visibility on an executive scale, and builds the support that makes this possible. [exec]
- Recognises when their own state is shaping the organisation's, and acts on themselves first. [exec]
- Takes proper recovery, a break without interruptions, and is seen to take it, because the organisation copies the top. [exec]
- Failure mode: going two years without a proper break and making decisions to end conversations. [exec]
- Failure mode: an anxious organisation whose anxiety began at the top, treated as everyone else's problem. [exec]
- Failure mode: a leader with nobody to tell. [exec]

7. Tensions

Availability versus judgment. The manager who is always available is valued, and makes worse decisions than the one who is not. Protected time and a visible boundary resolve it, and the discipline lies in holding to them when the urge to help pulls the other way.

Absorbing versus passing on. A good manager shields the team by taking load on themselves, until they break and the team gets all of it at once. The judgment lies in when to ask for help, which is before the state check comes back yes.

Boundaries versus the moment. Some weeks need the evenings and some crises need the weekend. The tension is between a time-boxed exception that ends when the time is up and a boundary that has quietly turned into a suggestion.

Self-care versus self-indulgence. A manager can read the recovery research as permission to do less, and a manager's load is real. The test is managerial leverage. A boundary that protects the manager's judgment on the things only they can do is a management practice. A boundary that protects them from the hard conversation is avoidance.

Honesty about state versus authority. Saying "I am overloaded and need help" can feel like an admission that will cost standing. The alternative costs more: snapping at people, cancelling one-on-ones and deciding badly, all of which people notice anyway.

8. Worked scenario

A manager promoted from within the team four months ago is still taking the two hardest technical tickets each sprint, because nobody else on the team has the context. They also attend the eight cross-team meetings their predecessor attended, and they do the management work (one-on-ones, planning, two hiring loops) in the evenings. They are visibly exhausted. They have cancelled one-on-ones twice in the past month, and this week they snapped at an engineer who asked a reasonable question. They have told nobody, because the team is delivering and asking for help feels like admitting they cannot do the job.

The temptation is to keep going. The team is delivering, the promotion is recent, and the evenings feel like the price of the transition. This is the failure the unit describes, and from inside it cannot be seen. The manager experiences full days and has no sense that their judgment has slipped. The cancelled one-on-ones and the sharp reply are the first evidence that it already has.

At a colleague's suggestion the manager keeps a time log. It shows what they half knew and had not looked at: twenty-two hours a week in meetings, most of which the predecessor attended out of habit, twelve hours of individual technical work, and the management done at night. Next comes the sort of recurring activities. The two hardest tickets are the most valuable thing to delegate, because the context only the manager holds is precisely what the team most needs to gain. The manager hands them to the two most senior engineers, with a brief and a pairing session each, and accepts that the tickets will take longer this sprint. Five of the eight meetings go. The manager declines three with a note and delegates two to an engineer who is closer to the subject. One-on-ones and hiring move into the working day, into the hours the meetings freed, and are fixed in the calendar.

Then the harder part. The manager tells their own manager what the past four months have been like, that they have been doing both jobs in the evenings, and that they are changing it. They ask for one thing: cover in the two cross-team meetings that do need a manager present, until the engineer taking them over is up to speed. They apologise to the engineer they snapped at and briefly explain why it happened. And they keep the boundary they have set, sending messages in the morning and not at midnight, where the team can see it.

The team notices the change before the manager does. The manager does not accept exhaustion as the price of the job. Had they done so, the team would have lost its manager and the manager would have lost the job.

9. Related competencies

- CC-4 Motivation, recognition, and sustainable pace: the team's pace and energy, for which the manager's own sets the example.
- PL-1 Delegation, ownership, and decision rights: delegation as the way a manager's time is freed, and Grove's point that a manager's output is the output of their organisation.
- PL-6 Leading through leaders: the leader of managers who notices an overloaded manager, and the manager who tells them.
- CC-1 Psychological safety and productive conflict: the safety that makes "I need help" possible to say, in both directions.
- PL-2 Feedback and difficult conversations: the apology for the short reply, as feedback owed to oneself as well as to the other person.

---

10. Self-check

1. Why is self-management a competency rather than a personal matter? *Answer:* A manager who cannot manage their own time, energy and boundaries cannot keep up any of the other competencies. One-on-ones are cancelled, the hard conversation is put off and decisions are made to end the meeting. The team also reads the manager's state and copies it.
2. What is the first practice, and why does the research say to do it? *Answer:* The time log, a two-week record of where the hours go. People are reliably wrong about where their time goes until they measure it, and the gap between the log and what the team needs is the first thing to change.
3. What decides what a manager keeps doing themselves? *Answer:* How much each activity yields, which turns on whether only they can do it. The things only they can do (a clear delegation, a hard conversation, a decision) are protected. Those others could do are delegated with a brief, declined or batched. Those that do not need doing are stopped.
4. What does the recovery research say a boundary is? *Answer:* Detachment, meaning psychological distance from work outside working time, which is the experience most strongly related to well-being. Stopping work is not enough on its own. A manager who answers messages every evening has not recovered, even if they have rested.
5. What is the honest position on decision fatigue? *Answer:* It is contested. The strong claim that self-control and decision quality run down over the day did not survive a large multi-laboratory replication. The advice that survives (make important decisions when rested, and do not decide something just to end a discomfort) rests on the recovery research and on experience, and does not depend on depletion.
6.

You realise you have been short with two managers this week and neither deserved it. What is it a signal of, and what do you do? [org] *Answer:* Of overload, and not of two problems with people. Treat it as the state check coming back yes: change the load that week by delegating something, declining something or asking for help, and apologise briefly to both, explaining why.

7. The organisation has become anxious and you notice it started with you. What is the repair? [exec] *Answer:* Act on yourself first. Take proper recovery and be seen to take it, lean on peers who can hear it, and show a visible change in your own state before sending any message to the organisation about theirs. The organisation copies the top.

Sources

- Peter F. Drucker, The Effective Executive (1967): know where your time goes; put first things first.
- Christina Maslach and Michael P. Leiter, Understanding the Burnout Experience: Recent Research and Its Implications for Psychiatry (2016): burnout as a condition of the job, applied to the manager.
- Sabine Sonnentag and Charlotte Fritz, The Recovery Experience Questionnaire: Development and Validation of a Measure for Assessing Recuperation and Unwinding from Work (2007): the four recovery experiences; detachment and well-being.
- Roy F. Baumeister, Ellen Bratslavsky, Mark Muraven and Dianne M. Tice, Ego Depletion: Is the Active Self a Limited Resource? (1998): the original ego-depletion claim.
- Martin S. Hagger, Nikos L. D. Chatzisarantis and 47 co-authors, A Multilab Preregistered Replication of the Ego-Depletion Effect (2016): the multi-laboratory replication that did not find it.
- Lara Hogan, Resilient Management (2019): the manager's own resilience and support.

Distributed and Hybrid Work

Why a chapter rather than a competency

Distributed and hybrid work is not a competency. A manager is not examined on it directly, and no question asks whether remote work is good. It is a condition under which most of the twenty-nine competencies are now exercised, and it pushes several of them the same way. Information arrives later and more filtered. Effort and contribution are harder to see. The boundary between work and the rest of life is thinner. Opportunity flows toward the centre unless something stops it. And the signals a manager relies on, such as the mood in the room or the question asked in a corridor, arrive faintly or not at all. This chapter states those changes once, so the competencies they affect can point here instead of each restating them. Those that point here most are CC-4 Motivation, recognition, and sustainable pace (motivation, recognition and pace), CC-3 Inclusive practices and equitable opportunity distribution (equitable opportunity), PL-5 Career development and growth paths (one-on-ones and growth), PL-2 Feedback and difficult conversations (feedback and the reaction to bad news), DE-5 Designing how work flows (how work flows) and SV-6 Organisational design and leading change (organisational design).

The default gradient

Left alone, a distributed organisation tilts toward wherever the decision-makers sit. Decisions are made in the meetings the centre can attend and information moves in the corridors the centre walks. The stretch work goes to the people the leader sees, and promotions follow. Nobody has to intend any of this, which is why reminders to include remote colleagues achieve nothing. A reminder does not counteract a gradient. Each design principle in this chapter is a way of engineering against the tilt, and each failure it describes is a way of leaving the tilt in place.

The gradient has been measured. In the randomised experiment reported in Nicholas Bloom, James Liang, John Roberts and Zhichun Jenny Ying, Does Working from Home Work? Evidence from a Chinese Experiment (2015), employees assigned to work from home were more productive and reported higher satisfaction than office-based colleagues doing the same job, yet they were promoted less often. Being out of sight took them out of the opportunity flow, even when they performed better. A later randomised trial, Nicholas Bloom, Ruobing Han and James Liang, Hybrid Working from Home Improves Retention without Damaging Performance (2024), tested a hybrid arrangement of two days at home and found no loss of performance or promotion, along with a substantial improvement in retention. Taken together, the two studies tell a manager what they need to know. The tilt is real and costs the organisation talent. It also comes from how the arrangement is designed

and is no law of nature: the hybrid design that kept people visible did not produce the penalty.

What changes, competency by competency

Information latency and filtering. A manager who used to hear about a problem the day it happened now hears about it in the next written update, after the person has decided how to present it. The instruments of PL-2 Feedback and difficult conversations and PL-6 Leading through leaders (skip-levels for signal, direct observation of artifacts, a visible reaction to bad news) matter more. The written record matters most. A distributed team runs on what is written, and a decision that lives only in a call was made without the half of the team that was not on it.

Visibility of effort and contribution. In an office, effort is seen. In a distributed team it has to be inferred from output, which favours the visible kinds of work and the people who narrate their own. Under CC-4 Motivation, recognition, and sustainable pace the manager has to go looking for what deserves recognition, since it will not be noticed in passing, and under CC-3 Inclusive practices and equitable opportunity distribution the opportunity ledger has to be kept by site, because the pattern will not show itself.

The boundary between work and life. When the desk is at home, the working day has no natural end, and a manager's late-evening message reads as an expectation. Sustainable pace under CC-4 Motivation, recognition, and sustainable pace and the manager's own boundaries under CC-6 Self-management become harder to hold and matter more, because the team takes its norm from what the manager visibly does.

The signals of disengagement. PL-5 Career development and growth paths treats withdrawal, cynicism and a halt in the flow of ideas as early signals of attrition, and across a video call they arrive later and fainter. The one-on-one is the instrument. The pattern shows up in the manager's notes across several of them.

Decision-making. A decision made in a meeting that one location cannot attend is a decision made for that location. The flow of work under DE-5 Designing how work flows and the structure under SV-6 Organisational design and leading change have to be designed so that decisions are location-neutral: made in writing, or in meetings every location can attend, and then published.

Onboarding. A new hire in an office absorbs context by proximity. A new hire at a distance absorbs what is written down and what a named guide tells them. For distributed hires the first-quarter plan under PL-3 Hiring and onboarding is mandatory, and the department baseline under CC-3 Inclusive practices and equitable opportunity distribution keeps the quality of onboarding from depending on which team someone happens to join.

Design principles

Engineering against the gradient looks like the principles below. They are written as design decisions because that is what they are. An attitude will not hold against a tilt.

1. **Written first.** Decisions, context and status are written down and published where every location can read them, and the written record is the canonical one. A meeting may come before the decision, but it does not replace the record.
2. **Location-neutral decision-making.** Anything that affects more than one location is decided by a process every location can take part in on equal terms: a proposal with a comment window, or a meeting at an hour every location can attend, recorded and summarised.
3. **Opportunity tracked by site.** The stretch work, the visible work, the strategic project and the promotions are counted by location as well as by person, and the pattern is corrected before it hardens.
4. **Rituals that work in every time zone.** Shared rituals are designed for the whole organisation instead of being run by one office and streamed to the rest. If one location experiences a ritual as a party and another as a broadcast, it is a belonging failure presented as a belonging initiative.
5. **Leaders assessed on their remote members' experience.** A manager with distributed reports is measured partly by asking those reports about their experience, because the default gradient favours the centre and the leader usually sits there.
6. **The manager's own boundary, visibly kept.** Messages go out in working hours, holidays are not interrupted, and the working day ends. The team learns its norm from what the manager does.
7. **Deliberate in-person time.** Where an organisation can bring distributed people together, it does so on purpose. The time goes on building relationships and having the hard conversations that a call handles badly. Status updates, which a document handles better, are a poor use of it.

Reading the scope tags in a distributed organisation

The scope tags (`[team]`, `[org]`, `[exec]`) apply unchanged. Distributed work widens the gap between scopes. How Judgment Scales describes information latency as growing with scope, and it grows faster when the layers also sit in different places, so the executive's structured sensing has to reach further. A manager of managers with teams in three cities is in effect working at slightly higher scope than the same manager with three teams along one corridor, and should read the `[org]` items with that in mind.

What does not change

The competencies themselves stay the same. A difficult conversation is still difficult over video, and a manager who defers it because the medium is awkward has still deferred it. Feedback is still specific, timely and anchored in behaviour. The intervention threshold is the same. The culture is still set in the costly moment, which people watch as closely on a screen as in a room. This chapter describes conditions. None of them is an exemption.

Techniques and Models

Every model and technique the units cite, one card each: what it is, its evidence grade, when to reach for it, and where it stops working. Grades: [research] peer-reviewed or large-sample empirical work; [practice] a widely adopted practitioner model with clear provenance; [contested] useful but disputed.

Leading people

Calibration [practice]

Aligning interviewers, and later managers, on what "meets the bar" means, so that the same capability or performance earns the same decision whoever assessed it. It rests on shared standards, on evidence counting for more than advocacy, and on a joint review of the evidence when two assessors disagree.

Use it when: Any hiring or promotion evaluation with more than one assessor; performance ratings across teams; reading leaders across organisations.

Limits: It decays into horse-trading as soon as ratings are negotiated instead of evidenced. If the most assertive manager's people are consistently rated highest, it has stopped being calibration.

Units: PL-3, PL-4, PL-5

Coaching managers through change [practice]

A leader supports the managers who report to them through their teams' transitions (growth, reorganisation, loss) with coaching questions in place of answers, and with sponsorship in place of protection.

Use it when: Whenever a manager who reports to you faces a situation they have not met before.

Limits: The whole difficulty is the line between coaching and rescuing. The leader crosses it the moment they take the decision themselves instead of developing the manager's.

Units: PL-6

Fair process [practice]

People accept a decision that goes against them when they were heard, the reasoning was explained, and what is expected next is unambiguous. The three elements are engagement, explanation and clarity of expectation.

Use it when: Any performance decision, promotion decision or selection that someone will be unhappy with.

Limits: Fair process cannot make a wrong decision right. What it does is get a defensible decision accepted. Performed as a ritual around a decision already made, it is recognised for what it is.

Units: PL-4

Feedback as a system [practice]

The mechanisms that replace the unfiltered signal a senior leader no longer gets in person. They include skip-level programmes, engagement data that somebody reads, calibrated reviews of leaders, and norms that make dissent at the leadership table safe.

Use it when: From organisational scope upward, whenever your picture of a team reaches you through its manager.

Limits: Every one of these mechanisms fails in the same way. It goes on existing on paper while the leader's visible reaction to bad news teaches people to stop using it.

Units: PL-2

The first ninety days [practice]

A new role treated as a transition with its own work (learning, early wins, the relationships the role depends on). It has explicit outcomes at thirty, sixty and ninety days, a first real contribution, a named guide and written context.

Use it when: Every hire, scaled to seniority. For senior hires, most of the ninety days goes on integration.

Limits: A plan is not a relationship. The framing fails without real support behind it, and when senior hires fail it is usually a failure of integration that no checklist would catch on its own.

Units: PL-3

Growth ladders and competency matrices [practice]

Explicit descriptions of what each level looks like for each competency. They make "what does the next level require?" a concrete question, and make promotion decisions comparable across people and teams.

Use it when: Career conversations; promotion decisions; fairness across teams; calibration.

Limits: A ladder describes what each level involves and sets no schedule. It decays into collecting checkboxes when levels are treated as tenure milestones instead of demonstrated judgment.

Units: PL-5

The halo effect [research]

One salient positive trait colours the assessment of unrelated traits. In hiring, the impressive credential, the strong first answer or the fluent presentation lifts every score that follows.

Use it when: Designing any evaluation. It is why each criterion is scored independently before discussion, and why debriefs start from the divergent scores instead of overall impressions.

Limits: Structure reduces the effect without removing it. A panel that resembles the manager brings it back as affinity.

Units: PL-3

Hiring as a system [practice]

At organisational scale, hiring has a throughput and a quality. It covers sourcing, a bar that survives growth, calibration across many interviewers, and the choice between internal and external candidates treated as a portfolio decision.

Use it when: Whenever more than a few people are hiring against the same bar, and whenever the organisation is growing fast.

Limits: A bar that drifts one interviewer at a time cannot be seen in any single hire and is obvious over a year. Taking the systems view helps only if someone looks at the year.

Units: PL-3

Levels of authority and intent-based leadership [practice]

Replace "may I..." with "I intend to...". The person states what they are about to do and why, the leader's silence counts as consent, and an objection has to be voiced. In its formal version, there is a ladder of authority levels agreed for each kind of decision: recommend, decide and inform, decide alone.

Use it when: Making the level of delegated authority explicit and visible in every exchange; moving decisions to where the information is.

Limits: It needs a leader who will in fact stay silent, and a culture in which stating an intent is not heard as asking permission.

Units: PL-1

Managerial leverage [practice]

A manager's output is the output of the organisation under them, and activities differ in how much output they produce per hour spent. An hour spent making a delegation clear pays back across every hour the delegate then spends on it.

Use it when: Deciding where a manager's time goes; explaining why a manager's performance is read from their team; making the economic case for explicit delegation.

Limits: Leverage can be negative, as when one hour of a manager's time interrupts ten people. The framing also says nothing about which outputs matter.

Units: PL-1, PL-4, CC-6

Managing managers [practice]

The move to leading managers, and the ways it typically goes wrong (skip-levels used badly, holding on to the old job, replacing a new manager who needed coaching). The people who report to you now need you to hold them accountable for their teams and to develop their judgment.

Use it when: The first year of leading managers, and whenever a leader notices they are doing a manager's job.

Limits: The feedback loop is longer and the evidence arrives filtered. Practices that worked when leading engineers give false confidence here.

Units: PL-6

The one-on-one as the report's meeting [practice]

The recurring activity with the highest leverage a manager has. It is protected and regular, the report sets much of the agenda, and the manager listens more than they talk. When the report is a manager, it becomes a judgment debrief.

Use it when: Every report, every week or fortnight, and never cancelled because of a need for status.

Limits: It turns into a status meeting as soon as the manager needs one. The early signs of disengagement show in the notes across several meetings, not in any single one.

Units: PL-5, PL-6

Radical candour [practice]

Two axes, caring personally and challenging directly, and the failures that follow when one is missing. Challenge without care reads as aggression. Care without challenge produces ruinous empathy, in which a well-liked manager's people never learn where they stand.

Use it when: Diagnosing why a manager who sees themselves as kind is not giving hard feedback.

Limits: It describes a stance and is no technique. It says why to have the conversation; situation–behaviour–impact says how.

Units: PL-2

Readiness [practice]

Someone is ready for a larger scope when they are already exercising that scope's judgment on real stakes, before they hold the title. Tenure, ambition and how much the person wants it do not count as readiness.

Use it when: Every promotion and succession decision, above all when the pull is toward whoever has waited longest or asked loudest.

Limits: The person must have been given the chance to show the judgment. A readiness rule without stretch assignments amounts to a rule that nobody is ever ready.

Units: PL-5, PL-6

Situation, behaviour, impact [practice]

Feedback anchored in a specific situation, describing observable behaviour instead of inferred character, and naming the impact. It removes the two triggers of defensiveness, generalisation and diagnosed intent, and gives the person something they can change.

Use it when: Any corrective or reinforcing feedback, at any seniority, in either direction.

Limits: The structure does not supply the courage. A well-shaped message that is never delivered is still avoidance. For a pattern, name the pattern and support it with several specific instances.

Units: PL-2

Situational leadership [practice]

Leadership style treated as a variable, adjusted (directing, coaching, supporting, delegating) to the person's competence and confidence on the task at hand.

Use it when: Calibrating how much structure to give anyone new to a task, including a new manager or a new executive.

Limits: Style has to follow the task, whatever the person's title. Misread it and you micromanage experts or abandon novices. Its predictive claims are thin, and it works as a vocabulary rather than a theory.

Units: PL-1, CC-2

Sizing teams and spans [practice]

How many people a manager can lead, and how many managers a leader of managers can support, treated as design constraints. The organisation's structure should follow them instead of following a tidy org chart. Span of control depends on the reports' maturity, the stability of the work and the coordination each report needs.

Use it when: Designing or changing structure; diagnosing an overloaded manager or an executive bottleneck.

Limits: The right numbers depend on context. A span that works in steady state fails during change, and optimising the number while ignoring the context gives neat charts and failing teams.

Units: PL-6, SV-6

Skill and will [practice]

Before responding to underperformance, separate cannot (a skill gap, met with teaching and support) from will not (a will gap, met with expectations and consequences). This standard adds the conditions gap, where the person was not given what the job required.

Use it when: The first question in every performance conversation.

Limits: The diagnosis is only as good as the evidence. Managers under pressure fall back on will, because that diagnosis asks less of them.

Units: PL-4

The Staff+ track [practice]

The senior individual-contributor path (staff, principal and beyond) as a career of equal standing to management. It has several distinct archetypes, and it recognises the coordinating and quality-raising work that otherwise goes unrewarded.

Use it when: Any career conversation with a strong engineer; any promotion process that has only a management path.

Limits: A track that appears on the ladder but not in the scopes and pay the organisation offers exists in name only. A manager who cannot describe it cannot offer it honestly.

Units: PL-5

Structured interviews [research]

Consistent questions mapped to a rubric defined before the process, scored independently before any discussion, by interviewers calibrated on the bar. Eighty-five years of selection

research place them among the most valid predictors of job performance, and they are fairer as well.

Use it when: Every hiring evaluation. The same discipline governs promotion decisions.

Limits: Structure reduces noise, and on its own it does not remove bias. A rubric carries its authors' assumptions; the remedy is deliberate design and a panel whose members do not all share them.

Units: PL-3

Task-relevant maturity [practice]

How much direction a person needs depends on their experience with this particular kind of work, and not on their general seniority. A senior engineer commanding their first incident, or a strong engineer newly promoted into management, has low maturity at the new task however senior they are.

Use it when: Setting delegation scope and level of support for each task and each person; calibrating support for a new manager.

Limits: Maturity has to be reassessed as tasks change. Yesterday's calibration quietly turns into today's mismatch.

Units: PL-1, PL-6, TJ-4

The three triggers [practice]

Why feedback is hard to receive. Truth triggers fire when it seems wrong, relationship triggers when the problem is who it comes from, and identity triggers when it threatens the receiver's sense of who they are. The giver can use this to anticipate the trigger, and the receiver to notice it firing.

Use it when: Shaping a hard message; receiving one.

Limits: It is a practitioner synthesis. Knowing the trigger does not defuse it; it only makes the reaction visible.

Units: PL-2

What good managers do [research]

Google's multi-year internal study of its own managers set out to test whether managers mattered and found that they did. It identified the behaviours that set the best apart: coaching, empowering instead of micromanaging, interest in people's wellbeing, a focus on results, communication, career support, a clear vision, and enough technical skill to help.

Use it when: Evidence that management behaviours can be measured and that the difference they make shows up in retention and satisfaction data.

Limits: It is a single-company study of one particular culture. The list holds no surprises, and its value lies in the evidence rather than in anything new.

Units: PL-6

Deciding

One-way and two-way doors [practice]

Classifying decisions by how reversible they are. Deliberate carefully on one-way doors, which are hard to reverse. Move fast on two-way doors, which are reversible, and let the people closest to a two-way door make the call.

Use it when: Allocating scrutiny; deciding which decisions to delegate; setting intervention thresholds; staging big bets so that more of them become reversible.

Limits: Reversibility is often misjudged, and "temporary" choices set hard. When in doubt, treat the door as one-way until it is proven otherwise.

Units: PL-1, DE-4, TJ-1, TJ-4

Running delivery

AI-assisted developer productivity [contested]

The effect of AI coding assistance on how fast and how well engineers work. A controlled experiment on a defined task found a large speed-up. A randomised trial with experienced developers working in their own mature repositories found a slowdown that the developers did not perceive. The effect depends on the work, the codebase and the engineer.

Use it when: Deciding whether and how to adopt AI assistance, and how to measure it. Measure in the team's own conditions, and never by self-report or by counting licences.

Limits: The evidence is recent and mixed, and the studies measure different things. The honest position is that the effect has to be measured locally. What is not contested is that the tools change what review must establish, and bring risks that need governance.

Units: DE-5, TJ-5, SV-5

Aligned autonomy [practice]

Direction, priorities and boundary standards set centrally, with methods left to teams. It answers what to standardise across teams (standardise where teams meet, and leave the inside to the team), and it is the balance an operating cadence strikes.

Use it when: Deciding what an organisation-wide standard should contain; any time more meetings are proposed as the answer to a failure of coordination.

Limits: Where "where teams meet" ends and "team internals" begin is a judgment, and it gets redrawn when a boundary fails. Drawn too wide, it becomes central control of methods. Drawn too narrow, it leaves the seams in chaos.

Units: DE-5, TJ-3

Batch size and predictability [practice]

Predictability comes from the delivery system rather than from effort. Smaller batches and limited work in progress reduce the variance in delivery times more reliably than pressure does.

Use it when: A team whose forecasts are wildly inconsistent; any programme trying to get predictability through exhortation.

Limits: Small batches need tooling that makes shipping them cheap. Without it, cutting batch size only multiplies overhead.

Units: DE-2

Blameless postmortems [practice]

Incident reviews that ask how the system allowed the failure and what to change (process, tooling, guardrails) instead of who to blame. This is the mechanism that makes honesty about failure safe, and so makes reliability improvable.

Use it when: After every significant incident or failed launch. It matters most when the stakes are highest and a senior person is involved, because that review is where the culture gets written.

Limits: Blameless does not mean free of consequences. Patterns of recklessness are performance matters and are handled separately. A review whose actions die in a backlog is a ritual and fixes nothing.

Units: DE-3, CC-1

Brooks's law [practice]

Adding people to a late software project makes it later, because the existing people must train the new ones and the communication paths multiply.

Use it when: Whenever "add people" is proposed for work that is already late; deciding whether more resourcing is a real option early enough to help.

Limits: It applies to work that cannot be partitioned. The judgment lies in telling that work apart from work that can.

Units: DE-2

The cone of uncertainty [practice]

Estimates are least accurate at the start of work, when least is known, and narrow as reality is discovered. Several practices follow: estimate in ranges, slice work small, de-risk the fuzziest part first, and re-forecast as the cone narrows.

Use it when: Setting expectations at the start of a project; explaining why an early date is a range in disguise.

Limits: The cone narrows only if someone does the discovering, and uncertainty nobody examines stays wide. Nor does the model excuse a failure to re-forecast once more is known.

Units: DE-2

Contract at the boundary [practice]

Making the agreement between a service and its consumers explicit, with an interface, a versioning and deprecation policy, and automated contract tests that catch a breaking change before integration does. It is the technical half of ownership and contract at the boundary.

Use it when: Teams that keep breaking each other's integrations; any widely consumed interface.

Limits: It verifies only the contract that was written down and cannot create interface discipline where there is none. At a boundary with a single consumer and no history of breakage, it is overhead.

Units: DE-4, TJ-3

Conway’s law [practice]

A system's structure mirrors the communication structure of the organisation that built it. As a diagnostic, chronic friction at the same boundary despite process fixes is often the structure presenting its bill. As a design rule, team boundaries and system boundaries are one decision.

Use it when: Reading recurring failures at a boundary; designing teams and systems together.

Limits: It is an observation with empirical support, and no law of nature. It says that the structures correlate, and does not say which one to change to fix the other.

Units: DE-4, TJ-1, SV-6

Cost of delay [practice]

The value lost for each unit of time a piece of work remains unshipped. Sequencing by cost of delay, rather than by effort, age or how loudly it is advocated, is the economic core of prioritisation.

Use it when: Ordering a queue; arguing a priority call; exposing the real price of "just one more thing first".

Limits: Most delay costs are estimates. Treat them as explicit assumptions to argue over, because false precision dresses opinion up as arithmetic.

Units: DE-1

Cycle time and historical forecasting [practice]

Forecasting from measurements of how long work in fact takes, instead of how long people hope it will take. It is the outside view put into operation, and history beats optimism.

Use it when: Forecasting; recalibrating people who chronically over- or under-commit; replacing estimation theatre.

Limits: History predicts only work that resembles the past, so novel work still needs spikes to de-risk it. A cycle-time distribution is a range, and quoting its average as a date repeats the original error.

Units: DE-2

Developer experience [practice]

Three dimensions that drive productivity as engineers experience it (feedback loops, cognitive load and flow state), all measurable and improvable. The framework distils its authors' earlier research; one of them, Nicole Forsgren, had led the DORA programme.

Use it when: Prioritising and reporting investment in the developer platform by the friction it removes rather than the tooling it installs.

Limits: The framework paper reports no new study, and its dimensions are measured by survey. It measures experience, which is not the same thing as delivered outcomes.

Units: DE-5

DORA stability metrics [research]

The research-validated delivery measures from the DORA programme, with deployment frequency and change lead time for throughput, and change failure rate and time to restore service for stability. The programme found that the organisations highest on throughput were also highest on stability.

Use it when: Assessing delivery health and tracking its trend, whether for a team against itself or across the organisation; the evidence that, at the level of the delivery system, reliability and speed do not trade off.

Limits: These are team-level measures of the system. They are never individual measures and never a leaderboard across teams. Once they become targets, Goodhart's law applies.

Units: DE-3, SV-5

Goodhart’s law [practice]

When a measure becomes a target, it ceases to be a good measure. It explains how healthy activity metrics can sit alongside flat outcomes.

Use it when: Any time a metric is about to be tied to a target, a reward or a comparison between teams.

Limits: The law names the failure without preventing it. The remedies (measure outcomes, use several measures, never rank people by them) are matters of judgment that the law itself does not supply.

Units: DE-5, SV-5

Ground truth and status criteria [practice]

Status defined by objective criteria instead of sentiment, stating what green means, what triggers red and what escalation each state demands. Reviews examine evidence instead of colour, and there is more than one channel to reality.

Use it when: From organisational scope upward, whenever the picture of a programme comes through its reports; whenever status smells wrong.

Limits: Criteria can be gamed like any other measure. The practice depends on visible safety for bad news; without it, the reporting system learns reassurance all over again.

Units: DE-4

Incident command [practice]

A defined coordination role during incidents, in which one person owns coordination and communication so that responders can focus on the fix. During an incident the priorities are to restore service, communicate, and then diagnose.

Use it when: Any incident that is more than trivial. Decide the role before the incident, because during one is too late. At organisational scale, incidents that cross teams need a commander with authority across those teams.

Limits: It is a coordination structure and offers no technical rescue. A commander who starts debugging has left the role.

Units: DE-3

The operating cadence [practice]

The organisation's rhythm of planning, review and escalation. A good one is light and rolling against stable priorities, with portfolio reviews based on evidence, explicit escalation triggers, and time set aside for correction. At scale, being responsive improves execution, and more detailed central planning does not.

Use it when: Quarterly planning that eats weeks and produces a stale plan; designing how an organisation senses and responds.

Limits: A rolling cadence is harder to explain to stakeholders who want the reassurance of a detailed plan, and it needs stable priorities to roll against.

Units: DE-5

The planning fallacy [research]

People predicting how long a task will take systematically underestimate it, even when they know similar tasks overran, because they reason from the case in front of them instead of from the distribution of similar cases. The corrective is the outside view, which forecasts from the reference class.

Use it when: Explaining why asking people to try harder does not fix estimation; the basis for historical forecasting.

Limits: A reference class has to exist and has to resemble the work. Novel work has none and needs de-risking instead.

Units: DE-2

Portfolio governance [practice]

An explicit, ranked and deliberately small set of concurrent initiatives at organisational scale, reviewed against evidence on a cadence. The ranking is used to settle contention for resources by strategy instead of by politics.

Use it when: Thirty priority-ones; scarce senior people fought over by sponsors; a failing initiative kept alive by its sponsor.

Limits: If the ranking is not published, or not used to settle the next contention, it decays back into the politics it replaced.

Units: DE-1

The pre-mortem [practice]

Before committing to a plan, assume it has failed and ask the team to write down why. This legitimises dissent and brings out the risks that optimism was suppressing, at far lower cost than the postmortem it prevents.

Use it when: The start of any significant initiative, especially where enthusiasm is high and disagreement has been quiet.

Limits: It produces candour only where safety already exists. In a low-trust room it produces the same silence as every other meeting.

Units: DE-4

Service-level objectives and error budgets [practice]

Define an explicit reliability target, and treat the tolerable shortfall against it (the error budget) as a resource to spend that gates risk-taking. With budget available, ship; with the budget exhausted, stabilise. The argument between reliability and velocity becomes a shared rule agreed in advance.

Use it when: Ship-or-stabilise conflicts that keep coming back; making "reliability first" operational instead of decorative.

Limits: It is only as good as leadership's willingness to obey it when that is inconvenient. Shipping through an exhausted budget in front of the organisation shows everyone what the real policy is.

Units: DE-3, CC-5

Team topologies and cognitive load [practice]

Teams have a limited cognitive load. The shape of a team and the ways it interacts with other teams are designed around that limit, and a platform team exists to reduce the load on the teams that ship product. How work flows is something to design.

Use it when: Designing or redesigning teams and their boundaries; deciding what a platform team is for.

Limits: It gives a design vocabulary and no formula. The four team types and three interaction modes describe options, and judging which one fits is still the manager's job.

Units: DE-5, SV-6

Trade-off transparency [practice]

Saying no by showing the cost of yes. State what the request costs in terms of what else moves, and let the person who wants it choose. The refusal becomes a decision the stakeholder owns, and the manager no longer quietly absorbs the demand.

Use it when: Every new request against a committed queue; every conversation in which someone asks for "just this one thing".

Limits: The cost has to be stated honestly. A cost inflated to make the answer no is a refusal in disguise, and stakeholders learn to discount it.

Units: DE-1, SV-2

Urgent versus important [practice]

Separating what demands attention now from what matters most, on the observation that urgency systematically crowds out importance unless it is deliberately resisted.

Use it when: Triaging demand on a team; diagnosing a calendar or a roadmap that is all reaction.

Limits: It is a lens and cannot schedule anything. Much real work is both urgent and important, and the discipline lies in funding the important but quiet work, which labelling quadrants does not do.

Units: DE-1

Work-in-progress limits and flow [practice]

Constraining how much is in flight so that the most important things finish sooner. A system loaded to full utilisation has unbounded queue times and no capacity to absorb variance. Starting less finishes more, and looking busy is a different thing from being effective.

Use it when: A team, department or portfolio where everything has been started and nothing ships; whenever "we need more people" might mean "we need less in flight".

Limits: Limits need enforcing at intake, or they become decoration. The slack they create has to be defended as capacity, and must not be harvested as if it were laziness.

Units: DE-1

Strategy and investment

Build, buy, or AI service [practice]

The three-way capability decision that v3.0 adds. Alongside building a capability or buying a product, there is the option of adopting an AI service that provides it. That third option has its own profile (low initial cost, uncertain ongoing cost, alternatives that go out of date quickly, dependence on a provider whose terms change), and it is weighed with the same questions as the other two.

Use it when: Any capability decision where an AI service is an option; the vendor register.

Limits: The AI option's cost and capability change faster than a decision can be revisited. The exit plan matters more than the entry price.

Units: SV-4

Communicating at the listener’s altitude [practice]

The same update cut again for each audience's concerns and level of abstraction (cost and risk, timeline and scope, impact and resolution), starting from what the listener has to decide.

Use it when: Every update to a non-engineering audience; every answer to vague executive pressure, which gets clarified before it is answered.

Limits: A message cut to the listener's altitude leaves out detail that matters to engineers. The judgment is about which details change the listener's decision, with the rest kept available instead of delivered.

Units: SV-2

The connecting thesis [practice]

The statement of why these teams do these things, tying several teams' work to a few department-level outcomes. It is what separates a strategy from a wish list. Without it you have the bottom-up roadmap.

Use it when: Reviewing any department roadmap; setting a manager of managers' own objectives at the altitude of outcomes.

Limits: A thesis that a team lead cannot recite governs nothing. It has to be communicated as a campaign, and a document alone will not do it.

Units: SV-1

Core versus context [practice]

Core is what differentiates a business in its customers' eyes. Context is everything else that must be done but does not differentiate. Build and invest in core; buy, adopt or outsource context.

Use it when: Any build, buy or partner decision, alongside total cost of ownership and, at executive scope, control and dependency risk.

Limits: The boundary moves, as yesterday's differentiator becomes a commodity. And "core" is a matter of differentiation, which is a different thing from importance.

Units: SV-4

Disagree and commit [practice]

When you disagree with a decision, say so directly and give your reasoning. If the decision goes the other way, commit to it fully instead of holding back effort while you wait to be proved right.

Use it when: A peer executive's strategy that you believe is wrong; any decision at the executive table that goes against you.

Limits: It presumes that a decision was in fact made and the disagreement was heard. Used to silence dissent before the decision, it becomes a different and worse rule.

Units: SV-2

Engineering as business partner [practice]

The stance that fixes "the department that says no". Bring options and trade-offs instead of vetoes, tie engineering work visibly to revenue and risk, make "yes, and here is the cost" the default grammar, and work with peers to change the system instead of fighting each instance.

Use it when: Engineering is seen as a cost centre; sales commits to custom features deal by deal; the board questions headcount.

Limits: It is about behaviour, and rhetoric will not carry it. It works only if the options are real and the costs honest, and indignation in their place loses the argument.

Units: SV-2

Explicit capacity allocation [practice]

A stated fraction of capacity for platform, reliability and debt, written down, agreed with the product partner and held under pressure. When it changes, it changes explicitly and for a stated period, and never by erosion.

Use it when: Every team's roadmap; the moment someone asks whether the platform work can wait just this once.

Limits: The number depends on context. The discipline is that a number exists and is defended. Zero is a decision with a date on which it fails.

Units: SV-3

The falsifiable scorecard [practice]

A small, stable set of measures declared when a strategy is set, covering outcomes, leading indicators chosen in advance, and trends in delivery-system and organisational health. It is reviewed on a cadence, revised along with the strategy, and able to prove the strategy wrong.

Use it when: Any strategy or major investment; the board's question of whether engineering is working.

Limits: It fails when it swells into a dashboard nobody reads, or when it is written after the results are in.

Units: SV-5

Headcount as investment [practice]

Engineering headcount and budget argued for as an investment with a return. The case covers what the money enables and protects and what its absence costs, it is staged with checkpoints, and it is defended with outcome measures instead of activity.

Use it when: Any budget request or defence; any cut, which should fall on the work that would not be started today.

Limits: Outcome measures have to exist. A budget defended by how busy everyone is has already lost.

Units: SV-4

Leading change [practice]

The eight-step model (urgency, a guiding coalition, a vision, communication, removing obstacles, short-term wins, consolidation, anchoring), used here as a checklist of what reorganisations leave out. Most often it is the last two steps.

Use it when: Any reorganisation, merger or transition; diagnosing why an earlier change did not stick.

Limits: It is prescriptive and based on cases. It works as a checklist of omissions and cannot guarantee that the change lands.

Units: SV-6, CC-5

Outcome versus output [practice]

Framing work as the change wanted in the world (outcome) instead of the thing to be built (output). Outcomes define success and leave the path open; outputs do neither. This is the central translation move of engineering strategy.

Use it when: Setting goals at every level; testing a roadmap item by asking why it is there; writing objectives that are not task lists.

Limits: Some work is legitimately shaped as output, such as compliance or contractual deliverables. And an outcome nobody can influence is a wish rather than a goal.

Units: SV-1

The SPACE framework [practice]

No single metric captures developer productivity. Five dimensions (satisfaction and wellbeing, performance, activity, communication and collaboration, efficiency and flow) are measured in balance. The framework serves as a check against one-dimensional measurement.

Use it when: Any proposed productivity metric. One that sits in a single dimension, and especially one that measures activity alone, is incomplete by construction.

Limits: It comes from a framework paper and is not an empirical finding. It says what to balance and gives no weighting.

Units: SV-5

Staged bets and checkpoints [practice]

A long investment structured to deliver intermediate value where it can, with its near-term cost stated honestly, commitment secured before it starts, and explicit checkpoints at which to double down or fold. This is how a two-year platform bet survives a quarter's urgency, and how a disruptive bet is sized so that it teaches something.

Use it when: Any multi-year platform investment; any bet on a disruptive technology; the existential priority that the organisation must fund by subtraction.

Limits: Staging can break up a bet whose value only arrives when it is whole. The judgment lies in finding the intermediate value without pretending it exists.

Units: SV-3, TJ-5

Strategy, vision, and roadmap [practice]

A vision names the destination and why it matters. A strategy is a diagnosis, a guiding policy and a coherent set of choices that includes what will not be done. A roadmap is the current sequence of bets that follows from it. A strategy that keeps every option open, or that everyone agrees with at once, has decided nothing.

Use it when: Writing or reviewing any strategy document; diagnosing a "strategy" that is in fact a wish list or a feature list.

Limits: The choices matter more than the documents. A brilliant strategy that nobody can recite governs nothing.

Units: SV-1

Structure follows strategy [practice]

Changes in strategy come before changes in structure and require them, and structures that outlive the strategy that produced them become the obstacle to the next one. The organisation is how the strategy is implemented.

Use it when: Any new strategy; checking that it has been implemented rather than only announced.

Limits: It is a historical thesis about large firms, and the direction of causation is disputed. The practical point is the corollary, that a strategy announced on top of the old structure has probably not been implemented.

Units: SV-6

Sunk-cost discipline [research]

Prior investment increases the tendency to continue a course of action regardless of its prospects. The discipline holds that what has been spent argues for nothing, and that only the value remaining against the cost remaining, compared with the alternatives, counts. It governs half-done migrations, championed bets and legacy systems.

Use it when: Any "we've come too far to stop" argument, especially when the bet is your own.

Limits: The opposite error also exists, which is abandoning sound long bets at the first fashionable alternative. Changed fundamentals justify switching; novelty does not.

Units: SV-3, SV-4

The three horizons [practice]

Balancing a portfolio across time between the current business, the emerging businesses that will matter in a few years, and the options that might disrupt everything. For an engineering roadmap, the executive question is how much capacity serves today's product, how much builds the platform the strategy needs, and how much explores.

Use it when: Roadmapping at executive scope; making the horizon mix an explicit decision instead of an accident.

Limits: The right proportions depend on context. The framework names the dimensions and leaves the answer to you.

Units: SV-3

Total cost of ownership [practice]

The full lifetime cost of a capability, meaning build cost plus maintenance forever, or purchase price plus integration, lock-in and exit. It is the corrective to every claim that something is "cheaper".

Use it when: Build-versus-buy, platform investment, vendor selection.

Limits: Long-horizon costs are estimates. The discipline is to name them explicitly so they can be argued over, without computing them to a false precision.

Units: SV-4

Writing the strategy down [practice]

A short written engineering strategy covering the diagnosis of the current situation, the policies that follow, the concrete actions and what it is not doing. It is reviewed on a cadence and revised when the signals say so, whatever the calendar says.

Use it when: Any organisation whose strategy exists only in conversation; the start of every planning cycle.

Limits: If the choices in it were never made, the document is no strategy. Writing it down tests the strategy; the page is not the product.

Units: SV-1

Technical stewardship

Architecture review forums [practice]

A standing forum where significant designs get cross-team scrutiny early. It is most valuable when it advises, teaches and catches cross-team consequences, and least valuable as a gate every change must pass.

Use it when: Keeping architecture coherent across autonomous teams; growing engineers' design judgment through exposure; reviewing the one-way doors.

Limits: Once it reviews everything, it becomes both a bottleneck and a rubber stamp. Scope it to decisions with cross-team or irreversible consequences.

Units: TJ-1

Code review as knowledge transfer [research]

Developers and managers expect review to find defects, yet its observed benefits are as much about knowledge transfer, team awareness and better solutions. Understanding the change is the main challenge reviewers face.

Use it when: Designing review as a way of teaching; the reason review must establish understanding, especially of generated code.

Limits: It is a single-company study. The proportions vary, and the structural finding holds.

Units: TJ-3

Compliance as a programme [practice]

A security or compliance mandate that touches many systems, led as a strategic effort with ownership, sequencing by risk, explicit capacity, honest roadmap impact and reporting. The alternative is spreading it across every team as a small tax.

Use it when: Any mandate that requires changing systems across the organisation.

Limits: It costs visible capacity, and that is deliberate. The alternative is a mandate squeezed into slack until the deadline is missed.

Units: TJ-6

Continuous incremental repayment [practice]

Improving the structure of code in small, behaviour-preserving steps as part of ordinary work, in the areas being changed anyway, plus a standing share of capacity for the highest-interest debt elsewhere. It replaces a separate rewrite.

Use it when: Every change to an area with known debt; the standing debt share; the answer to "later".

Limits: Repaying debt where you touch the code only repays what is touched. Debt in the areas everyone avoids needs the standing share, aimed deliberately.

Units: TJ-2

Debt as interest [practice]

Shipping expedient code is like borrowing. It lets you move faster now, and you pay interest on every later change until the principal is repaid. The metaphor turns expedience from a one-off into a running cost.

Use it when: Explaining debt to anyone; pricing it in terms of delivery.

Limits: It invites the wrong argument. Not all debt should be repaid, and the metaphor says nothing about which.

Units: TJ-2

Elevation to business risk [practice]

Turning technical risk into business terms and getting it decided at the level it deserves, so that platform debt becomes continuity risk and a vulnerability class becomes exposure. It comes with priced options and an honest roadmap impact, and replaces negotiating capacity team by team.

Use it when: Debt or exposure that threatens the business while product resists any slowdown.

Limits: Elevation does not mean escalating every item. It is kept for risks that are existential, or it turns into noise.

Units: TJ-2

The executive docket [practice]

An explicit, short list of the technical decisions an executive owns personally because they cannot be separated from company strategy. These are bet-the-company technology, build-versus-buy for the differentiating capability, lock-in of a core capability, foundational infrastructure, and the security and reliability posture. Everything else is delegated with real authority.

Use it when: Deciding which technical decisions belong to the executive; taking back one that is about to be made by default.

Limits: A docket that is too long is micromanagement with a title. Without one, the strategic decisions are made by whoever felt strongly.

Units: TJ-4

Installing a decision process [practice]

The usual way a manager of managers intervenes in a cross-team technical decision. Before giving any opinion, they set out the criteria, the evidence required, a named decider and a date, so that the decision is better and the organisation keeps the capability.

Use it when: A decision that crosses team boundaries with material consequences, cannot be reversed at department scale, is deadlocked, or quietly commits the whole department.

Limits: It is slower than giving the answer. In a real emergency, giving the answer is the intervention.

Units: TJ-4

The intervention threshold [practice]

Step in when being wrong would be costly and hard to reverse (one-way doors, safety, security, data integrity, customer impact in flight). Stay out when the decision is reversible and owning the outcome teaches more than the correction would. Intervene by raising what was missed instead of by overriding.

Use it when: Every technical decision heading somewhere the manager would not have taken it.

Limits: It fails in two mirror-image ways: overriding on preference, and avoiding a dangerous call to dodge conflict. It only works when applied consistently, whatever the manager's mood.

Units: TJ-4

The ironies of automation [practice]

When a task is automated, the human is left with the parts the automation cannot do, which are the hardest. The skills needed to intervene when it fails decay, because nobody practises them any more. And the operator ends up monitoring a system they understand less and less.

Use it when: Reasoning about what AI assistance does to an engineering organisation's understanding of its own systems; the case for reviewing for understanding and for building capability.

Limits: It is an analytical essay about industrial control and does not study software, so it applies here by analogy.

Units: TJ-5

The mirroring hypothesis [research]

The empirical test of Conway's observation across paired software products. Products built by loosely coupled organisations were significantly more modular than functionally equivalent products built by tightly coupled ones.

Use it when: The evidence that the team boundary and the system boundary are one decision in fact, and more than a metaphor.

Limits: It establishes the correlation, and does not say which structure to change to fix the other.

Units: TJ-1

Privacy by design [practice]

Privacy as the default, built into the design instead of bolted on, with data minimised and its handling visible. In practice it lives in everyday decisions about what is collected, where it is copied, who can read it, what is logged, and how deletion works across systems.

Use it when: Any system handling personal data; the design review; the access review.

Limits: It is a set of principles and offers no procedure. A deletion request that finds data in six places is what happens when the principles were not applied one decision at a time.

Units: TJ-6

Proportionate AI governance [practice]

An explicit, short policy for AI-assisted work. It sets what the tools may be used for, what data may be sent where, what verification and disclosure each class of change requires, and who owns a change whatever produced it. It is scaled to the risk of the path, reviewed as the tools change, and enforced in the pipeline where possible.

Use it when: Before or at adoption. Across teams, it is set once at the seams and the choice of tools is left to teams.

Limits: It fails in two mirror-image ways: a blanket prohibition, which people ignore, and no policy at all, which is discovered in an incident.

Units: TJ-5

Quality as a system [practice]

Tests, continuous integration, guardrails, standards and the paved road working as a system that makes the right thing the easy thing. The manager invests in the system instead of inspecting change by change. Coverage is no measure of quality; tests that catch what matters are.

Use it when: Any quality problem; the manager who is tempted to review every change.

Limits: A system is only as good as its enforcement at the seams. A definition of done that nobody enforces where work crosses teams is just a document.

Units: TJ-3

Review for AI-generated code [practice]

What review must establish when an assistant produced some or all of a change. The submitter understands it and can explain it; it is tested at the level the risk warrants; its origin is disclosed where required; it is checked against the vulnerability classes that generated code is prone to; and it imports no unacceptable licence. The standard is set before adoption and is proportionate to the change's risk.

Use it when: Every team using AI assistance, enforced in the pipeline where it can be.

Limits: A review standard that tries to keep pace with volume stops establishing understanding. This one holds review to understanding and lets throughput be whatever it turns out to be.

Units: TJ-3

Risk-based triage [practice]

Prioritising security findings by exploitability (whether the vulnerable path is reachable) and exposure (what it gives an attacker, and to how many), and fixing the whole class where several findings share a cause.

Use it when: A scanner's four hundred findings; an audit's list; any count that has paralysed a team.

Limits: Exploitability is a judgment about reachability, and it changes as the system changes. Any triage is a snapshot.

Units: TJ-6

Security of AI-generated code [research]

Across security-relevant scenarios, a substantial share of assistant-generated code contained known classes of vulnerability. Tools and models change the proportion, and the structural point stands. Generated code has a vulnerability profile that is checked as a class, and is not to be assumed safe because it reads fluently.

Use it when: The checklist for generated code; scanning aimed at the known failure modes on the paths where a vulnerability would cost most.

Limits: It is a point-in-time study of one tool. The rate is not stable, and the class-level risk is.

Units: TJ-3, TJ-5, TJ-6

The strangler pattern [practice]

Replacing a legacy system incrementally. New capability grows around the old system and traffic is routed away piece by piece, until the old system can be retired. It is the standing alternative to the big-bang rewrite, and the reason a manager can insist that the incremental path is explored first.

Use it when: Any proposal to rewrite a working system from scratch.

Limits: It takes longer to reach a clean end state, and it needs interfaces in the old system to strangle at, which some systems lack.

Units: TJ-1

Systemic risk signals [practice]

The signals no single team can see, such as incident trends across teams, lengthening lead times, change-failure rates in particular areas, the module everyone avoids, and the one expert who alone understands a system. They are read and acted on at organisational scope before a team's own metrics force the issue.

Use it when: From organisational scope upward, as the manager of managers' own instruments.

Limits: Signals get read only if someone looks. The failure is waiting for a team to ask.

Units: TJ-2

Systems understanding over currency [practice]

The knowledge that matters for leading is the map of the system (its shape, risks and dependencies, and who understands each part). A manager keeps the map by reading designs and code without becoming an approver, staying in the room for hard decisions, and rotating deep dives. Depth in the code is a different thing, and trying to keep it while managing usually fails at both jobs.

Use it when: Every manager who has stopped coding daily; every executive tempted to intervene in a stack that has moved on.

Limits: It rests on experience. The failure it describes, expertise frozen at the last hands-on year, is easy to recognise, but the map decays too if the reading stops.

Units: TJ-4

The technical-debt quadrant [practice]

Debt is deliberate or inadvertent, and prudent or reckless. Deliberate, prudent debt is a trade taken knowingly. Reckless debt is mess with a flattering name. Inadvertent, prudent debt is what learning ordinarily produces.

Use it when: Classifying debt before arguing about it; setting repayment priority by interest rate.

Limits: Classifying debt changes nothing on its own. Managing it means the tracking, priced impact and scheduled repayment built on top.

Units: TJ-2

Threat modelling [practice]

Four questions asked of a system (what are we building, what can go wrong, what are we going to do about it, did we do a good job) that make security a design activity instead of a scan at the end.

Use it when: The design review for any system handling personal or financial data; questions a manager can ask without being a security engineer.

Limits: A threat model is a snapshot. Systems and threats move on, and a model nobody revisits describes a system that no longer exists.

Units: TJ-6

Shaping culture

Artifacts, espoused values, and assumptions [practice]

Culture at three levels. There are the visible artifacts, the values the organisation espouses, and the underlying assumptions that drive behaviour in practice, which are usually unspoken and often at odds with the espoused values. Where they diverge, the behaviour is the culture.

Use it when: Diagnosing an organisation's real culture; naming the assumption that a costly moment revealed.

Limits: It diagnoses and does not prescribe. It says where to look, and is silent on how to change what you find.

Units: CC-5

Autonomy, mastery, and purpose [research]

Intrinsic motivation, which drives quality and persistence on complex work, depends on autonomy, competence and relatedness ("purpose" in the popular gloss), and controlling external rewards undermine it. Pay and perks prevent dissatisfaction without creating engagement.

Use it when: Designing roles; diagnosing disengagement; resisting the reflex to fix morale with benefits.

Limits: It assumes fair pay as the floor, and purpose is no substitute for salary. Chronic overwork also burns through all three drivers regardless.

Units: CC-4

Burnout as a job condition [research]

Burnout (exhaustion, cynicism, reduced efficacy) is caused by the conditions of the job, namely workload, control, reward, community, fairness and values. A shortage of individual resilience is not the cause.

Use it when: Any burnout, whether the team's or the manager's own. It is why the fix goes to the cause by changing the load, and why wellness programmes laid over an impossible plan are theatre.

Limits: The six conditions are broad. The diagnostic work of finding which one is producing the burnout still falls to the manager.

Units: CC-4, CC-6

Coaching the coaches [practice]

Developing managers' judgment by coaching their reasoning on real situations in the judgment debrief. The manager's own patterns of avoidance are treated as the development need, and verdicts are withheld when a manager asks for one.

Use it when: Every one-on-one with a manager who reports to you; building a coaching culture among managers.

Limits: It works only if the leader models it. A manager of managers who gives commands cannot mandate a coaching culture.

Units: CC-2

Conflict then commitment [practice]

Trust makes productive conflict possible, and productive conflict produces commitment. The matching failures are artificial harmony, where conflict is avoided, and ambiguity, where commitment was never reached. Disagreement belongs before the decision and stops after it.

Use it when: Closing decisions explicitly; diagnosing a team that never argues, or one that never stops.

Limits: It is a practitioner model told as a fable. Its sequence is a useful vocabulary and has not been established as a finding.

Units: CC-1

The costly moment [practice]

A situation in which a stated value and the pressure of the moment point in different directions, and people are watching the decision. What the leader does then is what the organisation learns. A value that holds only when it costs nothing is decoration.

Use it when: Reading culture; deciding on a launch against an exhausted error budget; the brilliant jerk; the shaded report.

Limits: It rests on experience. People recognise it so readily that nearly every scenario in the competency is built from one.

Units: CC-5

Culture as what is rewarded, tolerated, and punished [practice]

The working definition this standard uses. Culture is the accumulated pattern of how people behave in practice, set by what leaders visibly do, reward, promote and tolerate, far more than by stated values, policies or announcements.

Use it when: Reading an organisation's real culture; planning culture change by realigning hiring, promotion, recognition and consequences.

Limits: It implies patience, because changing culture means realigning over several quarters. It also implies fragility, because one visible exception can undo quarters of work.

Units: CC-5

Distributed and hybrid work [practice]

The conditions under which most competencies are now exercised. Effort is less visible, the line between work and life is thinner, disengagement signals arrive later, and opportunity tilts toward the centre. The design principles here engineer against that tilt. The Distributed and Hybrid Work chapter covers this in full.

Use it when: Recognition and pace on a distributed team; the opportunity ledger by site; location-neutral decisions.

Limits: These are conditions and grant no exemptions. The competencies stay the same, and a hard conversation put off because the medium is awkward has still been put off.

Units: CC-4

Diversity and decision quality [contested]

The formal argument that groups with diverse perspectives outperform groups of uniformly able people on complex problems, together with the widely cited consultancy studies that link workforce diversity to financial performance. The formal argument depends on its assumptions, and the consultancy findings have not survived independent re-analysis.

Use it when: The make-up of the rooms where decisions are taken, treated as a matter of decision quality. Homogeneous groups demonstrably converge fast, and can converge on the wrong answer.

Limits: The strong quantitative claims are not established. The equity case does not rest on them, and a case that overclaims is easier to dismiss.

Units: CC-3

Ego depletion and decision fatigue [contested]

The claim that self-control and decision quality draw on a limited resource that runs down through the day, so that later decisions are worse. A large preregistered multi-laboratory replication did not find the effect.

Use it when: As a caution, since managers should not plan their days around a resource that may not exist. The advice that survives (decide important things when rested, and do not decide just to end a discomfort) rests on the recovery research and on experience.

Limits: The original literature is large and the replication is a single one, if a strong one. The honest position is that the strong claim is unsupported.

Units: CC-6

The five keys of team effectiveness [research]

Google's internal study of what set its effective teams apart. Psychological safety was the strongest of five differentiators, ahead of dependability, structure and clarity, meaning, and impact.

Use it when: Confirming that, in a large engineering organisation with its own data, the construct from the academic literature was the one that mattered most.

Limits: It is a single-company study reported by the company itself. Its value is as confirmation, and it discovered nothing new.

Units: CC-1

Glue work [practice]

The essential, under-recognised work that makes teams function (coordination, documentation, onboarding, unblocking), which tends to be done by people who are then not promoted for it. Alongside it sits office housework: notes, scheduling, cover.

Use it when: Auditing who does what; designing recognition and promotion criteria that do not punish the people holding the team together.

Limits: The fix is to value the work and share it out, and eliminating it is no fix. A team with no glue does not become efficient. It gets stuck.

Units: CC-3

The GROW model [practice]

A coaching structure (Goal, Reality, Options, Will) that gives "ask before telling" a shape a manager can hold in a one-on-one without training.

Use it when: Any developmental conversation; the default alternative to supplying the answer.

Limits: It structures a conversation. It cannot replace judgment about when a conversation is the wrong tool.

Units: CC-2

The learning organisation [practice]

An organisation designed to turn experience into shared capability. Incident learnings travel between teams, postmortem actions are audited to completion, slack is funded for learning, and leadership reviews ask what was learned as seriously as they ask what was shipped.

Use it when: Executive scope; building the learning loops and the coaching cascade into the institution.

Limits: The book's claims about systems thinking go further than what is used here. A learning culture that exists only in policy dies in the first executive review that looks for someone to blame.

Units: CC-2

The manager’s own resilience [practice]

The manager's own state through the changes their teams go through. That means noticing one's own reaction, building a support network made up of people other than one's reports, and asking for help. At executive scope it also means the deliberate support that makes sustained judgment possible.

Use it when: Any period of sustained load; the leader who has no one to tell.

Limits: It is practical advice, and not clinical. A manager whose exhaustion has become burnout needs the load changed; practising resilience will not do it.

Units: CC-6

Managing your own time [practice]

Record your time to learn where it goes. Cut what does not need doing, and consolidate what remains into usable blocks. Put first things first, which means deciding what not to do. People are reliably wrong about where their time goes until they measure it.

Use it when: The time log; the leverage sort; a calendar that is all meetings.

Limits: It was written for executives in a different era. The principles transfer better than the specific prescriptions.

Units: CC-6

Merger integration with cultural intent [practice]

Avoid both forced assimilation, where the acquirer's culture wins by default and the acquired talent leaves, and indefinite coexistence. Understand both cultures honestly, define the shared identity and the non-negotiables deliberately, integrate with respect and explicit intent, and keep the best of both.

Use it when: Any merger or acquisition; the executive's defining equity case.

Limits: It covers the cultural half of integration. The structural half (which platform, which teams, on what timeline) belongs to organisational design.

Units: CC-3

Opportunity and attention as scarce resources [practice]

The two resources a manager hands out that shape careers. One is the visible work that helps people grow; the other is mentorship, airtime, credit and the benefit of the doubt. Both flow along a default gradient: to the similar, the confident, the vocal and the near.

Use it when: The opportunity ledger; turning "be inclusive" into a question a manager can audit and act on.

Limits: It is ordinary observation. Its power is that it makes the distribution visible, and the visible pattern is usually different from the one the manager believed in.

Units: CC-3

Proximity and promotion [research]

In a randomised experiment, employees assigned to work from home were more productive and more satisfied, and were promoted less often than office-based colleagues doing the same job. A later randomised trial of a hybrid arrangement found no promotion penalty and better retention.

Use it when: Evidence that the default gradient is real, costs the organisation talent, and is a property of design and not a law.

Limits: These are two studies in specific settings. Full-time home working carried a promotion penalty and the visible hybrid design did not, so the effect depends on how the arrangement is designed.

Units: CC-3

Psychological safety [research]

A belief shared by the members of a team that the team is safe for interpersonal risk-taking, such as admitting error, asking questions and challenging. It predicts team learning and performance. It is independent of standards, and high safety combined with high standards is the learning zone.

Use it when: Always. It is the precondition for every honest signal this standard depends on: bad news, dissent, blameless reviews, upward feedback.

Limits: It is a different thing from niceness, comfort or an absence of conflict. Measured badly, it flatters itself, and in an organisation that is unsafe even anonymous surveys read too high.

Units: CC-1

Recognition as feedback [practice]

Recognition follows the rules of feedback (specific, timely, anchored in behaviour) and adds one of its own. It is traced to contribution instead of visibility, and at organisational scope it rewards what the organisation wants in practice, including shared and invisible work.

Use it when: Every moment of recognition; designing a recognition system across teams.

Limits: Recognising only individuals who hit their goals teaches people to drop the glue work. The system has to be designed to prevent that.

Units: CC-4

Recovery and detachment [research]

Recovery from work depends on specific experiences (psychological detachment from work during non-work time, relaxation, mastery, control over one's time), and detachment in particular is the one most strongly related to well-being in the study that defined them (a cross-sectional study, so it shows association, not cause). A boundary is defined by the presence of detachment, which is a different thing from the absence of work.

Use it when: The manager's own boundaries; the team's; the evening message that is read as an expectation.

Limits: It measures experiences by self-report. The mechanism is well supported, and designing the prescriptions is left to the manager.

Units: CC-6

The rescue reflex [practice]

The manager's instinct to step in with the solution the moment someone struggles. It feels like help and is faster today, and it teaches the person they cannot be trusted to think. In an organisation it takes the form of doing a manager's job; at executive level, of an organisation that waits for the executive's answer.

Use it when: Every wobble; the question of who decides at the end of the conversation.

Limits: It rests on experience. The judgment lies in the counter-practice, which is to support or coach, and to rescue only when being wrong would be costly and irreversible, and then to do it explicitly.

Units: CC-2

Sustainable pace [practice]

A team works at a pace it can keep up indefinitely, because overtime is borrowed from next week's productivity and the loan compounds. Occasional, time-boxed crunch for a real reason may be acceptable. Chronic crunch is a management failure.

Use it when: The pace watch; the launch that tempts the team's own enthusiasm; praise for the crunch.

Limits: It is a practitioner principle. Its evidence is every team that has tried the alternative for more than a few weeks.

Units: CC-4

Trust as a behavioural account [practice]

Trust in leadership works like an account into which only behaviour makes deposits. Telling the truth consistently when it is costly, following through visibly, admitting mistakes and explaining decisions (the hated ones included) all pay in. Inconsistency, spin and broken promises withdraw. A campaign of reassurance is a withdrawal that looks like a deposit.

Use it when: Falling trust in a survey; the hostile narrative after a reorganisation; carrying out painful decisions with integrity.

Limits: It describes trust and cannot measure it. Other people read the account, and the leader is usually the last to see the balance.

Units: CC-1

Glossary

- **Accountability without fear**: Clear ownership and honest consequences for patterns of unaccountability, combined with real safety for honest error. People must know exactly which of the two applies to them. Conflate them and both are lost. (CC-1, PL-4)
- **Admission criteria**: The written statement of how an initiative gets onto a department roadmap (expected contribution to stated outcomes, cost, risk, dependencies), so that a request is answered with the criteria rather than with a no. (SV-3)
- **Affinity bias**: Favouring people who resemble oneself in evaluation. It is why interview panels should not resemble the manager, and one of the things structured interviewing is designed to counter. (PL-3, CC-3)
- **AI governance**: The explicit, proportionate policy for AI-assisted work. It says what the tools may be used for, what data may be sent where, what verification and disclosure each class of change requires, and who owns a change whatever produced it. It is scaled to the risk of the path and enforced in the pipeline where it can be. (TJ-5, TJ-3)
- **Aligned autonomy**: Direction, priorities and boundary standards set centrally, with methods left to teams. This is the balance an operating cadence strikes at organisational scale, and the answer to the question of what to standardise across teams. (DE-5, TJ-3)
- **Architecture review forum**: A standing forum where significant designs get cross-team scrutiny early. It is most valuable when it advises, teaches and catches cross-team consequences, and least valuable as a gate every change must pass. (TJ-1, TJ-4)
- **Blameless postmortem**: An incident review that asks how the system allowed the failure and what to change, rather than who to blame. It makes honesty about failure safe, and so makes reliability improvable. Blamelessness does not mean an absence of accountability, which is handled separately. (DE-3, CC-1)
- **Bottom-up roadmap**: A department "strategy" assembled from team wish lists, with no connecting thesis. More detail in the list never fixes it. The fix is to define the outcomes and derive the list again from them. (SV-1)
- **Brilliant jerk**: A high performer whose behaviour breaches the team's values. Tolerating one teaches everyone that results buy exemption. (PL-4, CC-5)
- **Brooks's law**: Adding people to a late software project makes it later, because the existing people must train the new ones and the communication paths multiply. It applies to work that cannot be partitioned. (DE-2)
- **Build, buy, or AI service**: The three-way capability decision (build it, buy a product, or adopt an AI service), with each option weighed by the same questions: is this core, what is the lifetime cost, and where are we left if the provider or partner changes course? (SV-4, TJ-5)

- **Burnout**: Exhaustion, cynicism and reduced efficacy caused by the conditions of the job (workload, control, reward, community, fairness, values) rather than by a shortage of individual resilience. It is treated at the cause, by changing the conditions. (CC-4, CC-6)
- **Bus factor**: The number of people who could leave before critical knowledge or capability is lost. It is a dependency to engineer away, and once the person has given notice it is an option with an expiry date. (TJ-2, TJ-3)
- **Calibration**: Aligning evaluation standards across interviewers, managers or organisations so that the same capability or performance earns the same judgment. Evidence counts over advocacy, and assessors who disagree review the case together. (PL-3, PL-4, PL-5)
- **Capacity allocation**: A stated share of a team's capacity for platform, reliability and debt, held under pressure, so that foundational work is a decision rather than whatever is left over. Zero is also a decision, and it comes with a date on which it fails. (SV-3)
- **Charter**: The explicit statement of what a team or department owns. Charter disputes are settled at the level that sets charters instead of being fought commit by commit. (SV-4, SV-6)
- **Coaching versus directing**: Developing someone's judgment through questions, as against supplying the answer. Coaching is the default for developmental moments. Directing is the named exception for emergencies, complete novices and stakes that leave no room. (CC-2)
- **Commitment versus estimate**: An estimate is the best forecast. A commitment is what is promised outward, deliberately buffered, with its scope, known risks and variance protocol stated. Saying both aloud makes the buffer visible and builds trust. (DE-2)
- **Conditions gap**: Underperformance caused by what the person was not given (clear requirements, access, time, a functioning team) rather than by a lack of skill or will. It is the third branch of the skill-and-will diagnostic, and the one managers most often miss. (PL-4)
- **Cone of uncertainty**: Estimates are least accurate at the start of work, when least is known, and narrow as reality is discovered. What follows from it: give ranges, cut small slices, de-risk the fuzziest part first, and re-forecast as the cone narrows. (DE-2)
- **Connecting thesis**: The statement of why these teams do these things, tying several teams' work to a few department-level outcomes. Without one, you have a bottom-up roadmap. (SV-1)
- **Contract testing**: Automated verification of the agreement between a service and its consumers, which catches a breaking change at the boundary before integration does. It is the technical half of ownership and contract at the boundary. (DE-4, TJ-3)
- **Core versus context**: Build what differentiates the business in its customers' eyes (core), and buy or adopt the commodity (context) so engineers are free for

differentiating work. Core means differentiating, which is a different thing from important. Email is important and almost never core. (SV-4)

- **Cost of delay**: The value lost for each unit of time a piece of work remains unshipped. Sequencing by cost of delay, rather than by effort, age or advocacy, is the economic core of prioritisation. (DE-1)
- **Costly moment**: A situation in which a stated value and the pressure of the moment point in different directions, and people are watching the decision. Culture is set in these moments. A value that holds only when it costs nothing is decoration. (CC-5, CC-1)
- **Culture**: The accumulated pattern of what behaviour gets rewarded, tolerated and punished in practice. What leaders visibly do sets it far more than stated values, policies or announcements. Where behaviour and the stated values diverge, the behaviour is the culture. (CC-5)
- **Cycle time**: How long work takes from start to done, as measured. It is the basis of historical forecasting, and it is a distribution rather than a single number. Quoting its average as a date repeats the original estimation error. (DE-2)
- **Decision process**: The usual way a manager of managers intervenes in a cross-team technical decision. Before any opinion is issued, they install the criteria, the evidence required, a named decider and a date, so that the organisation keeps the capability. (TJ-4, TJ-1)
- **Decision rights**: The explicit statement of who may decide what, and at which level of authority. Where more than one person could reasonably claim a decision, the rights are designed and written down in advance instead of being discovered during the dispute. (PL-1, SV-6)
- **Default gradient**: The tilt of a distributed organisation toward wherever its decision-makers sit. Decisions, information, stretch work and promotions flow to the centre without anyone intending it. Design counters it; reminders never do. (DHW, CC-3)
- **Delegation brief**: The written form of a delegation for anything larger than a day's work: outcome, constraints, authority level, who else is involved, and when the work is next reviewed together. If the brief cannot be written, the delegation is not ready. (PL-1)
- **Developer experience**: The friction between an engineer and a shipped change (feedback loops, cognitive load, flow state), treated as measurable capacity. Investment in it is judged by the friction removed rather than the tooling installed. (DE-5)
- **Disagree and commit**: Argue against a decision directly and with reasoning. If it goes the other way, commit to it fully instead of holding back effort while waiting to be proved right. This presumes that a decision was in fact made and that the disagreement was heard. (SV-2)
-

Disengagement signals: Withdrawal, cynicism and ideas that stop coming. A manager who is paying attention sees these early signs months before a resignation. On a distributed team they are fainter and arrive later. (CC-4, PL-5)

- **DORA metrics**: The four research-validated delivery measures (deployment frequency, lead time for changes, change failure rate, time to restore service). Together they describe a team's delivery system, and they move together in high-performing organisations. They are team-level measures, never individual ones, and never a leaderboard. (SV-5, DE-3)
- **Ego depletion**: The contested claim that self-control and decision quality draw on a limited resource that runs down through the day. A large multi-laboratory replication did not find the effect. The practical advice that survives rests on the recovery research instead. (CC-6)
- **Engineering strategy**: The written translation of business goals into technical direction. It contains a diagnosis, a set of choices that includes what will not be done, and the reasoning. It is reviewed against reality and revised when signals call for it, not because the calendar does. (SV-1)
- **Error budget**: The tolerable shortfall against a service-level objective, treated as a resource to spend that gates risk-taking. With budget available, ship; with the budget exhausted, stabilise. It is only as good as leadership's willingness to obey it when that is inconvenient. (DE-3, CC-5)
- **Executive docket**: The explicit, short list of technical decisions an executive owns personally because they cannot be separated from company strategy. These are bet-the-company technology, build-versus-buy for the differentiating capability, lock-in of a core capability, foundational infrastructure, and the security and reliability posture. Everything else is delegated with real authority. (TJ-4)
- **Fair process**: A decision people accept even when it goes against them, because they were heard, the reasoning was explained, and what is expected next is unambiguous. Its three elements are engagement, explanation and clarity of expectation. (PL-4)
- **Falsifiable scorecard**: A small, stable set of measures declared when a strategy is set, able to show that the strategy is failing. It covers the outcomes the strategy was meant to move, leading indicators chosen in advance, and trends in delivery and organisational health. Without one, success gets judged by the effort spent. (SV-5)
- **First ninety days**: Treating a new role as a transition with its own work (learning, an early real contribution, the relationships the role depends on), with explicit outcomes at thirty, sixty and ninety days. For senior hires, most of the ninety days goes on integration. (PL-3)
- **Glue work**: The coordinating, documenting and unblocking work that holds a team together and rarely shows up in the artifacts a promotion process values. The people who do it tend not to be promoted for it. (PL-5, CC-3)

- **Goodhart's law**: When a measure becomes a target, it ceases to be a good measure. It explains how healthy activity metrics can sit alongside flat outcomes, and it is the caution attached to every metric in this standard. (DE-5, SV-5)
- **Ground truth**: The real state of a piece of work, established by looking rather than by reading a status report. Above team scope, managing risk is largely a matter of managing the reporting system that carries status upward. (DE-4)
- **GROW model**: A coaching structure (Goal, Reality, Options, Will) that gives "ask before telling" a shape a manager can hold in a one-on-one. (CC-2)
- **Growth ladder**: An explicit description of what each level looks like for each competency, so that a promotion is judged against a shared description rather than against how assertive a manager is. Also called a competency matrix. It describes what each level involves, and sets no schedule. (PL-5)
- **Halo effect**: One salient positive trait colouring the assessment of unrelated traits. In hiring, it is the impressive credential or strong first answer that lifts every later score. Independent scoring per criterion, before any discussion, counters it. (PL-3)
- **Horizon mix**: The explicit split of an organisation's capacity between serving the current business, building the platform the strategy will need in a few years, and exploring what might disrupt everything. It is a decision owned at executive level, and should never be a residual. (SV-3)
- **Improvement plan**: A formal, time-boxed, supported chance to meet a stated bar, with the outcome at the end defined when the plan is set. It is fair only if the person already knew the bar and had been given feedback. Paperwork for a decision already made does not count as one. (PL-4)
- **Incident command**: A single defined role, assigned before the incident, that owns coordination and communication so that responders can focus on the fix. A commander who starts debugging has left the role. (DE-3)
- **Innovation budget**: An explicit, bounded allocation for evaluating new technology, with pilots that have criteria, a contained blast radius and an honest review. Adoption is then neither prohibited nor stampeded, and the team that took the risk is not left holding the liability. (TJ-3)
- **Intent-based leadership**: Replacing "may I..." with "I intend to...". The person states what they are about to do and why, the leader's silence counts as consent, and an objection has to be voiced. Decisions move to where the information is, and the leader stays informed. (PL-1)
- **Interface discipline**: Versioning, a deprecation policy, contract tests and communicated change for a widely used interface. The owning team's definition of done includes its consumers' stability. (TJ-3, DE-4)

- **Intervention threshold**: Step into a technical decision when being wrong would be costly and hard to reverse (one-way doors, safety, security, data integrity, customer impact in flight). Stay out when it is reversible and owning the outcome teaches more than the correction would. Apply the threshold consistently, and intervene by raising what was missed rather than by overriding. (TJ-4)
- **Inverse Conway manoeuvre**: Designing the organisation to produce the architecture you want, on the basis that the system will mirror the organisation's structure either way. (SV-6, TJ-1)
- **Judgment debrief**: The form a one-on-one with a manager takes. You walk through the hard call they face or have just made, probe the reasoning, and coach the thinking instead of issuing a ruling. This is how managers develop. (PL-6)
- **Key-person risk**: Critical capability, context or ownership concentrated in one person, so that the organisation is destabilised when they leave or are courted. It is a structural property to design away rather than a personal failing to lament. (PL-5, PL-6, DE-4)
- **Leading change**: What a structural change needs beyond its announcement. That means the why, what changes and what stays, the effect on every leader whose scope moves, presence through the transition, and consolidation so the change sticks. (SV-6)
- **Learning organisation**: An organisation designed to turn experience into shared capability. Incident learnings travel between teams, postmortem actions are audited to completion, slack is funded for learning, and leadership reviews ask what was learned. (CC-2)
- **Learning zone**: The combination of high psychological safety and high standards. High safety with low standards is the comfort zone. Low safety with high standards produces silence and hidden failure. (CC-1)
- **Location-neutral**: Decision-making and information flows designed so that a team four time zones away has the same access as one down the corridor. In practice that means proposals with comment windows, meetings every location can attend, and decisions published in writing. (DHW, CC-3)
- **Managerial leverage**: Grove's framing that a manager's output is the output of the organisation under them, and that activities differ in how much output they produce per hour spent. An hour spent making a delegation clear pays back across every hour the delegate then spends on it. (PL-1, PL-4, PL-5)
- **Mirroring hypothesis**: The empirically tested form of Conway's observation. Products built by loosely coupled organisations are measurably more modular than equivalent products built by tightly coupled ones. (TJ-1)
- **Office housework**: Necessary, low-visibility tasks such as notes, scheduling and cover, which fall along predictable and inequitable lines unless they are distributed deliberately. (CC-3)
-

One-on-one: The recurring, protected meeting between a manager and a report. It belongs to the report and is not for status. Blockers, frustrations, growth and the earliest signs of disengagement come up here. With a manager as the report, it becomes a judgment debrief. (PL-5, PL-6)

- **One-way door**: A decision that is hard to reverse and so deserves deliberation and senior involvement. A two-way door, by contrast, is reversible and should be decided quickly by the people closest to it. When in doubt, treat a door as one-way until shown otherwise. (PL-1, TJ-1, TJ-4)
- **Operating cadence**: An organisation's rhythm of planning, review and escalation. It is sound when it is light and rolling against stable priorities, reviews evidence, has explicit escalation triggers and leaves time for correction. When it does none of that, it is planning theatre. (DE-5)
- **Outcome versus output**: Framing work as the change wanted in the world instead of the thing to be built. Outcomes define success and leave the path open; outputs do neither. The test is to ask, of any item in a plan, why it is there. (SV-1)
- **Ownership map**: The written record of every system, capability and decision and who owns each, checked before and after any structural change. Whatever has no owner on the map is where the next incident will come from. (SV-6, DE-3)
- **Paved road**: A well-supported default platform and toolchain that makes the good path the easy one. Teams may leave it if they take on the ownership cost themselves. A paved road nobody would choose is a mandate in costume. (TJ-3)
- **Planning fallacy**: The systematic tendency to underestimate how long a task will take, even when similar tasks are known to have overrun, because the forecast reasons from the case at hand instead of from the distribution of similar cases. The outside view corrects it. (DE-2)
- **Portfolio governance**: An explicit, ranked and deliberately small set of concurrent initiatives at organisational scale, reviewed against evidence on a cadence. The ranking settles contention for resources by strategy instead of by politics. (DE-1)
- **Pre-mortem**: Before committing to a plan, assume it has failed and ask the team to write down why. This legitimises dissent and brings out the risks that optimism had suppressed. It works only where safety already exists. (DE-4)
- **Privacy by design**: Privacy as the default, built into ordinary engineering decisions (what is collected, where it is copied, who can read it, what is logged, how deletion works in practice) instead of bolted on afterwards. (TJ-6)
- **Proximity bias**: Opportunity and promotion going to the people a leader sees, regardless of performance. It was measured directly in the work-from-home experiment, where remote employees outperformed and were promoted less. (CC-3, DHW)
-

Psychological safety: A shared belief that the environment is safe for interpersonal risk-taking, such as admitting error, asking questions and challenging. It is one of the strongest predictors of team performance. It is independent of standards, and it is different from comfort or an absence of conflict. (CC-1)

- **Reaching around**: A leader of managers overriding a manager's decisions with that manager's own engineers, fixing their team's problems directly, or letting their escalations become the leader's workload. It is the organisational form of taking delegated work back. (PL-1, PL-6)
- **Readiness**: Already exercising the judgment of a larger scope on real stakes before holding the title. Tenure, ambition and how much the person wants the role are no measure of it. (PL-5, PL-6)
- **Recovery**: The experiences through which people recuperate from work (psychological detachment, relaxation, mastery, control over one's time). Of these, detachment is the one most strongly related to well-being. A boundary is defined by the presence of detachment, which is a different thing from the absence of work. (CC-6)
- **Rescuing**: Stepping in with the solution the moment someone struggles. It feels like help and is faster today, and it teaches people they cannot be trusted to think. It is the anti-pattern of both coaching and delegation. (CC-2, PL-1)
- **Risk-based triage**: Prioritising security findings by exploitability (whether the vulnerable path is reachable) and exposure (what it gives an attacker, and to how many), and fixing the whole class where several findings share a cause. It answers both failure modes, paralysis at the count and ignoring it. (TJ-6)
- **Self-management**: The manager's judgment applied to the manager. It covers where their time and energy go, what boundaries they hold, and how they stay able to think clearly under load. It is the competency that sustains all the others. (CC-6)
- **Service-level objective**: An explicit reliability target for a service, from which the error budget is derived. It settles how reliable is reliable enough, so the argument is not re-fought for every decision. (DE-3)
- **Service ownership**: Every service has a named owner responsible for its operation, its interface stability, its documentation and its future. A critical service with no owner comes first, because everything about its health depends on somebody being responsible. (TJ-3, DE-3)
- **Situation, behaviour, impact**: A feedback structure that names the specific situation, describes the observable behaviour instead of inferred character, and states the impact. It removes the two triggers of defensiveness, generalisation and diagnosed intent. (PL-2)
- **Situational leadership**: Adjusting leadership style (directing, coaching, supporting, delegating) to a person's competence and confidence on the task at hand. The style follows the task, whatever the person's title. (PL-1)

- **Skill and will**: The first diagnostic question about underperformance is whether the person cannot or will not, because the right responses are entirely different. This standard adds a third branch, the conditions gap. (PL-4)
- **Skip-level**: A regular one-on-one between a leader and the people who report to their reports. It exists for unfiltered signal about team health and to make the leader visibly accessible. It is never a channel for decisions that belong to the manager. (PL-2, PL-6)
- **Span of control**: How many reports a leader can serve effectively. There is no constant. It depends on the reports' maturity, the stability of the work, and how much coordination each report needs. A span that works in steady state fails during change. (PL-6, SV-6)
- **Staff+ track**: The senior individual-contributor career path (staff, principal and beyond), of equal standing to management and with several distinct archetypes. A manager who cannot describe it cannot offer it honestly. (PL-5)
- **Strangler pattern**: Replacing a legacy system incrementally by growing the new capability around it and routing traffic away piece by piece until the old system can be retired. It is the standing alternative to the big-bang rewrite. (TJ-1, TJ-2)
- **Structure follows strategy**: Changes in strategy require changes in structure, and structures that outlive the strategy that produced them become the obstacle to the next one. The organisation is how the strategy is implemented. (SV-6, SV-1)
- **Structured interview**: Consistent questions mapped to a rubric defined before the process starts, scored independently by each interviewer before any discussion. It is among the best-evidenced predictors of job performance, and fairer as well. (PL-3)
- **Structured sensing**: How an executive keeps their technical judgment current. The means are architecture and operations reviews with evidence, principal engineers who tell the truth, periodic deep dives into the systems carrying the most risk, and samples of raw signal that keep the aggregates honest. (TJ-4)
- **Succession**: A ready or nearly ready successor for every key role, the executive's own included, treated as an explicit expectation of senior roles and not as a personal virtue. It is the structural answer to key-person risk. (PL-5, PL-6)
- **Sunk cost**: What has already been spent, which argues for nothing in a decision about the future. Only the value remaining against the cost remaining, compared with the alternatives, counts. Changed fundamentals justify switching a bet; novelty does not. (SV-4, SV-3)
- **Sustainable pace**: A pace a team can hold indefinitely. Occasional, time-boxed crunch for a real reason may be acceptable. Chronic crunch is a management failure, and a run of heroics is a failure of the system rather than a badge. (CC-4)
- **Systems understanding**: The knowledge that lasts after a manager stops coding and matters for leading, namely the system's shape, risks and dependencies and who

understands each part. A manager keeps it by reading designs without approving them, staying in the hard rooms, and rotating deep dives. (TJ-4)

- **Task-relevant maturity**: How much direction a person needs on a specific kind of work, set by their experience with that work and not by their general seniority. It is reassessed as tasks change. (PL-1, PL-6)
- **Technical debt**: The implied cost of future rework from choosing expedience now. A controlled amount is a sensible trade. Left unmanaged, the interest (slower delivery, more defects, more toil) compounds until movement stops. (TJ-2)
- **Technical-debt quadrant**: Debt classified as deliberate or inadvertent, and as prudent or reckless. Deliberate, prudent debt is a trade taken knowingly; reckless debt is mess with a flattering name. The quadrant gives a team the vocabulary to classify debt before arguing about it. (TJ-2)
- **Threat modelling**: A design discipline built on four questions (what are we building, what can go wrong, what will we do about it, did we do a good job). It makes security part of design instead of a scan at the end, and a manager can ask the questions without being a security engineer. (TJ-6)
- **Time log**: Two weeks of recording every hour as it is spent, then comparing the record with what the team needs. It is the first practice of self-management, because people are reliably wrong about where their time goes until they measure it. (CC-6)
- **Total cost of ownership**: The full lifetime cost of a capability, meaning build cost plus maintenance forever, or purchase price plus integration, lock-in and exit. It corrects reasoning from the sticker price. The discipline lies in naming the costs explicitly, not in computing them precisely. (SV-4)
- **Trade-off transparency**: Saying no by showing the cost of yes ("we can do this, and that slips; which do you want?"), so the stakeholder makes an informed choice and the manager does not quietly absorb the demand. (DE-1)
- **Watermelon status**: Reporting that is green on the outside and red on the inside. It points to a broken information system that has learned to optimise for reassurance, rather than to a broken team. The fix is objective status criteria, evidence-based review and visible safety for bad news. (DE-4)
- **Work-in-progress limit**: A cap on how much is in flight at once, so that the most important things finish sooner. Starting less finishes more. Unless the limit is enforced at intake, it is decoration. (DE-1)

Sources

- John Allspaw. *Blameless PostMortems and a Just Culture*. Etsy, Code as Craft (engineering blog), 2012. — the case for blameless incident reviews; the distinction between blamelessness and the absence of accountability
- Hal R. Arkes and Catherine Blumer. *The Psychology of Sunk Cost*. Organizational Behavior and Human Decision Processes, 35(1), 124–140, 1985. — prior investment increases the tendency to continue a course of action independent of its prospects
- Alberto Bacchelli and Christian Bird. *Expectations, Outcomes, and Challenges of Modern Code Review*. Proceedings of the 35th International Conference on Software Engineering (ICSE), 712–721, 2013. — review's benefits are knowledge transfer and awareness as much as defect finding; understanding the change is the central challenge
- Mehrdad Baghai, Stephen Coley and David White. *The Alchemy of Growth: Practical Insights for Building the Enduring Enterprise*. Perseus Books, 1999. — the three horizons of a growth portfolio
- Lisanne Bainbridge. *Ironies of Automation*. Automatica, 19(6), 775–779, 1983. — automation leaves the human the hardest parts and erodes the skills needed to intervene
- Roy F. Baumeister, Ellen Bratslavsky, Mark Muraven and Dianne M. Tice. *Ego Depletion: Is the Active Self a Limited Resource?*. Journal of Personality and Social Psychology, 74(5), 1252–1265, 1998. — the original ego-depletion claim
- Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 1999. — sustainable pace (originally the forty-hour week)
- Betsy Beyer, Chris Jones, Jennifer Petoff and Niall Richard Murphy (editors). *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media, 2016. — service-level objectives and error budgets; incident management; postmortem culture
- Jeff Bezos. *2015 Letter to Shareholders (Amazon.com, Inc.)*. Amazon.com, Inc., 2016. — the "one-way door / two-way door" classification of decisions by reversibility
- Jeff Bezos. *2016 Letter to Shareholders (Amazon.com, Inc.)*. Amazon.com, Inc., 2017. — disagree and commit
- Nicholas Bloom, James Liang, John Roberts and Zhichun Jenny Ying. *Does Working from Home Work? Evidence from a Chinese Experiment*. Quarterly Journal of Economics, 130(1), 165–218, 2015. — randomised experiment: higher productivity and satisfaction at home, and a promotion penalty

SOURCES

- Nicholas Bloom, Ruobing Han and James Liang. *Hybrid Working from Home Improves Retention without Damaging Performance*. Nature, 630, 920–925, 2024. — randomised trial of two days at home: no performance or promotion penalty, substantially better retention
- Frederick P. Brooks Jr. *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, 1975. — adding people to a late software project makes it later
- Ann Cavoukian. *Privacy by Design: The 7 Foundational Principles*. Information and Privacy Commissioner of Ontario, 2009. — privacy as the default, embedded in design
- Alfred D. Chandler Jr. *Strategy and Structure: Chapters in the History of the American Industrial Enterprise*. MIT Press, 1962. — structure follows strategy: changes in strategy require changes in structure
- Melvin E. Conway. *How Do Committees Invent?*. Datamation, April 1968, 1968. — the original statement that a system's structure mirrors the communication structure of the organisation that built it
- Stephen R. Covey. *The 7 Habits of Highly Effective People*. Free Press, 1989. — the urgent/important matrix as popularly framed
- Ward Cunningham. *The WyCash Portfolio Management System*. OOPSLA '92 Experience Report, ACM, 1992. — the origin of the technical-debt metaphor
- Peter F. Drucker. *The Effective Executive*. Harper & Row, 1967. — know where your time goes; put first things first
- Amy C. Edmondson. *Psychological Safety and Learning Behavior in Work Teams-*. Administrative Science Quarterly, 44(2), 350–383, 1999. — psychological safety defined and measured; its link to team learning
- Amy C. Edmondson. *The Fearless Organization: Creating Psychological Safety in the Workplace for Learning, Innovation, and Growth*. Wiley, 2018. — safety and standards as independent axes; the learning zone
- Nicole Forsgren, Jez Humble and Gene Kim. *Accelerate: The Science of Lean Software and DevOps*. IT Revolution Press, 2018. — the DORA research programme; throughput and stability improve together; the four delivery metrics
- Nicole Forsgren, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck and Jenna Butler. *The SPACE of Developer Productivity: There's More to It Than You Think*. ACM Queue, 19(1), 2021. — five dimensions of developer productivity; no single metric suffices
- Camille Fournier. *The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change*. O'Reilly Media, 2017. — the management career arc level by level; managing managers; ladders
- Martin Fowler. *TechnicalDebtQuadrant*. martinfowler.com, 2009. — debt as deliberate or inadvertent, prudent or reckless

SOURCES

- Martin Fowler. *Refactoring: Improving the Design of Existing Code (Second Edition)*. Addison-Wesley, 2018. — small behaviour-preserving improvements as ordinary work
- Martin Fowler. *StranglerFigApplication*. martinfowler.com, 2004. — incremental replacement of a legacy system by growing the new one around it
- David A. Garvin. *How Google Sold Its Engineers on Management*. Harvard Business Review, December 2013, 2013. — the account of Google's internal study of its managers (Project Oxygen) and the behaviours that distinguished the best
- Jeremiah Green and John R. M. Hand. *McKinsey's Diversity Matters/Delivers/Wins Results Revisited*. Econ Journal Watch, 21(1), 2024. — re-analysis finding the consultancy diversity–performance results do not replicate
- Andrew S. Grove. *High Output Management*. Random House, 1983. — task-relevant maturity; managerial leverage; the one-on-one; the manager's output as the output of their organisation
- Martin S. Hagger, Nikos L. D. Chatzisarantis and 47 co-authors. *A Multilab Preregistered Replication of the Ego-Depletion Effect*. Perspectives on Psychological Science, 11(4), 546–573, 2016. — the multi-laboratory replication that did not find the effect
- Paul Hersey and Kenneth H. Blanchard. *Management of Organizational Behavior: Utilizing Human Resources*. Prentice-Hall, 1969. — the origin of situational leadership: style flexed to the follower's readiness on the task
- Lara Hogan. *Resilient Management*. A Book Apart, 2019. — coaching managers and teams through change; sponsorship; the manager's own resilience
- Daniel Kahneman and Amos Tversky. *Intuitive Prediction: Biases and Corrective Procedures*. TIMS Studies in Management Science, 12, 313–327, 1979. — the planning fallacy and the outside view (reference-class forecasting)
- W. Chan Kim and Renée Mauborgne. *Fair Process: Managing in the Knowledge Economy*. Harvard Business Review, July–August 1997, 1997. — engagement, explanation and expectation clarity as the elements of a process people accept
- Gary Klein. *Performing a Project Premortem*. Harvard Business Review, September 2007, 2007. — the pre-mortem technique
- John P. Kotter. *Leading Change*. Harvard Business School Press, 1996. — the eight-step model of organisational change
- Max Landsberg. *The Tao of Coaching: Boost Your Effectiveness at Work by Inspiring and Developing Those Around You*. HarperCollins, 1996. — the skill/will diagnostic and matrix
- Will Larson. *An Elegant Puzzle: Systems of Engineering Management*. Stripe Press, 2019. — organisational design and systems thinking at scale; sizing teams and managers' spans; hiring as a system

SOURCES

- Will Larson. *The Engineering Executive's Primer: Impactful Technical Leadership*. O'Reilly Media, 2024. — writing an engineering strategy; the executive's operating rhythm
- Will Larson. *Staff Engineer: Leadership Beyond the Management Track*. Stripe Press, 2021. — the senior individual-contributor track and its archetypes
- Patrick Lencioni. *The Five Dysfunctions of a Team: A Leadership Fable*. Jossey-Bass, 2002. — trust, conflict and commitment as a sequence; artificial harmony
- Alan MacCormack, Carliss Baldwin and John Rusnak. *Exploring the Duality between Product and Organizational Architectures: The Mirroring Hypothesis*. Research Policy, 41(8), 1309–1324, 2012. — empirical test of Conway's observation: loosely coupled organisations produced more modular products
- L. David Marquet. *Turn the Ship Around!: A True Story of Turning Followers into Leaders*. Portfolio / Penguin, 2012. — intent-based leadership: "I intend to..." as a mechanism for pushing authority to where the information is
- Christina Maslach and Michael P. Leiter. *Understanding the Burnout Experience: Recent Research and Its Implications for Psychiatry*. World Psychiatry, 15(2), 103–111, 2016. — burnout as exhaustion, cynicism and reduced efficacy, caused by job conditions
- Steve McConnell. *Software Estimation: Demystifying the Black Art*. Microsoft Press, 2006. — the cone of uncertainty in its standard software form; ranges; decomposition
- Joel Becker, Nate Rush, Elizabeth Barnes and David Rein (METR). *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*. METR (Model Evaluation and Threat Research), July 2025, 2025. — randomised controlled trial; experienced developers in their own repositories were slower with AI tools while believing they were faster
- Geoffrey A. Moore. *Dealing with Darwin: How Great Companies Innovate at Every Phase of Their Evolution*. Portfolio, 2005. — core versus context
- Abi Noda, Margaret-Anne Storey, Nicole Forsgren and Michaela Greiler. *DevEx: What Actually Drives Productivity*. ACM Queue, 21(2), 2023. — the developer-experience framework: feedback loops, cognitive load, flow state
- Scott E. Page. *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies*. Princeton University Press, 2007. — the formal argument that diverse heuristics outperform uniform ability on complex problems
- Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt and Ramesh Karri. *Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions*. IEEE Symposium on Security and Privacy, 2022. — a substantial share of assistant-generated code in security-relevant scenarios contained known vulnerability classes
-

SOURCES

Sida Peng, Eirini Kalliamvakou, Peter Cihon and Mert Demirer. *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot*. arXiv preprint 2302.06590, 2023. — controlled experiment; a defined task completed substantially faster with an AI coding assistant

- Daniel H. Pink. *Drive: The Surprising Truth About What Motivates Us*. Riverhead Books, 2009. — autonomy, mastery and purpose as the popular synthesis of self-determination research
- Tanya Reilly. *The Staff Engineer's Path: A Guide for Individual Contributors Navigating Growth and Change*. O'Reilly Media, 2022. — senior individual-contributor leadership; the source of "glue work"
- Donald G. Reinertsen. *The Principles of Product Development Flow: Second Generation Lean Product Development*. Celeritas Publishing, 2009. — cost of delay; queues, utilisation and work in progress; batch size and variance
- Julia Rozovsky. *The Five Keys to a Successful Google Team*. Google re:Work, 2015. — Google's internal study of team effectiveness: psychological safety the strongest differentiator
- Richard P. Rumelt. *Good Strategy Bad Strategy: The Difference and Why It Matters-*. Crown Business, 2011. — the kernel of good strategy: diagnosis, guiding policy, coherent action; the hallmarks of bad strategy
- Richard M. Ryan and Edward L. Deci. *Self-Determination Theory and the Facilitation of Intrinsic Motivation, Social Development, and Well-Being*. American Psychologist, 55(1), 68–78, 2000. — autonomy, competence and relatedness as the needs behind intrinsic motivation
- Edgar H. Schein. *Organizational Culture and Leadership*. Jossey-Bass, 1985. — artifacts, espoused values and underlying assumptions
- Frank L. Schmidt and John E. Hunter. *The Validity and Utility of Selection Methods in Personnel Psychology: Practical and Theoretical Implications of 85 Years of Research Findings*. Psychological Bulletin, 124(2), 262–274, 1998. — meta-analysis of selection-method validity; structured interviews among the strongest predictors
- Kim Scott. *Radical Candor: Be a Kick-Ass Boss Without Losing Your Humanity*. St. Martin's Press, 2017. — care personally, challenge directly; ruinous empathy
- Peter M. Senge. *The Fifth Discipline: The Art and Practice of the Learning Organization*. Doubleday, 1990. — the learning organisation
- Adam Shostack. *Threat Modeling: Designing for Security*. Wiley, 2014. — threat modelling as a design discipline; the four questions
- Matthew Skelton and Manuel Pais. *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. IT Revolution Press, 2019. — team shapes and interaction modes; cognitive load; the platform team; Conway's law in practice

SOURCES

- Sabine Sonnentag and Charlotte Fritz. *The Recovery Experience Questionnaire: Development and Validation of a Measure for Assessing Recuperation and Unwinding from Work*. Journal of Occupational Health Psychology, 12(3), 204–221, 2007. — detachment, relaxation, mastery and control as recovery experiences
- Douglas Stone and Sheila Heen. *Thanks for the Feedback: The Science and Art of Receiving Feedback Well*. Viking, 2014. — truth, relationship and identity triggers; receiving feedback
- Marilyn Strathern. *'Improving Ratings': Audit in the British University System*. European Review, 5(3), 305–321, 1997. — the standard phrasing of Goodhart's law: when a measure becomes a target, it ceases to be a good measure
- Edward L. Thorndike. *A Constant Error in Psychological Ratings*. Journal of Applied Psychology, 4(1), 25–29, 1920. — the original description of the halo effect
- Michael D. Watkins. *The First 90 Days: Critical Success Strategies for New Leaders at All Levels*. Harvard Business School Press, 2003. — the transition into a new role as its own work; early wins; integration
- Sloan R. Weitzel. *Feedback That Works: How to Build and Deliver Your Message-*. Center for Creative Leadership Press, 2000. — the situation–behaviour–impact feedback structure
- John Whitmore. *Coaching for Performance: GROWing People, Performance and Purpose*. Nicholas Brealey, 1992. — the GROW model

Changelog

EMA Standard v3.0

Released and examined from 26 September 2026. Every sitting started on or after that day draws from the v3.0 question bank and its credential records framework version 3.0. From 27 October 2026 a sitting also draws its questions by the per-competency weights in its Exam Content Outline.

Body of Knowledge v3.1, 7 October 2026

A minor version under the governance rules: clarified prose, and the corrections of 8 October listed below.

The prose of every knowledge unit, the five domain hubs, the index, the glossary definitions, the technique cards and the chapters on how judgment scales, distributed work, governance and citation was edited for plainness. Sentences are shorter and more direct, and the habits that made the text read as if generated (stacked dashes, reflexive contrasts, slogan endings) are gone. Every claim, qualification, example, number, named model, evidence grade and source was meant to stay unchanged in meaning; the corrections below put back the few places where it did not. Nothing was added and nothing substantive was removed.

What did not change:

- Every anchor. Each section, principle, model, tension and self-check question keeps its number, each principle and tension keeps its bold lead-in word for word, and each model keeps its name. The sentences after a lead-in were reworded with the same meaning, so a citation such as *EMA Body of Knowledge v3.0, PL-1.P3* points at the same principle, saying the same thing, in v3.1.
- The Competency Framework and the Exam Content Outlines. They are what is examined, a minor version may not change them, and they stay at v3.0. What a credential records is the framework version, so credentials are unaffected.
- The bibliography, the glossary terms and the technique cards' names and references. One evidence grade changed, as a correction listed below.

The v3.0 text of the Body of Knowledge is not kept as a separate archive: v3.1 says the same things in plainer words.

Corrections to v3.1, 8 October 2026

A review of v3.1 against v3.0 found sentences where the plainer wording had moved the meaning. They now say what v3.0 said:

- TJ-2 Technical debt: after a major incident the executive commits *to the board* a remediation programme. The rewrite had the executive committing the board.
- PL-1 Delegation, ownership, and decision rights: in the worked scenario the failure lies in the delegation, not in the delegate; and task-relevant maturity depends on experience with the task rather than on general seniority.
- PL-4 Performance management: a leader's bad quarter with an outside cause is judged on their decisions, not on the raw outcome. PL-6 Leading through leaders: leading leaders is a different job, not a larger version of leading engineers. PL-2 Feedback and difficult conversations: the upward-feedback question asks about the manager's own behaviour.
- TJ-6 Security, privacy, and compliance as engineering responsibilities principle 4: the fix goes into what produces the pattern, not into today's instances. DE-3 Reliability, incident response, and operational ownership: a rising on-call load is not the team's private problem. DE-2 Estimation, planning, and honest commitments: the competency is not producing accurate estimates.
- Firm statements the rewrite had softened, restored in DE-1 Prioritisation under constraint, DE-2 Estimation, planning, and honest commitments, DE-5 Designing how work flows, CC-1 Psychological safety and productive conflict, CC-2 Coaching versus directing, CC-3 Inclusive practices and equitable opportunity distribution, CC-4 Motivation, recognition, and sustainable pace, the People Leadership hub, the glossary, three technique cards and *Distributed and Hybrid Work* (where the 2015 study's finding is again self-reported satisfaction).

The same review checked three sources, cited in four places, against the original publications. The attributions predate v3.1:

- Sabine Sonnentag and Charlotte Fritz, The Recovery Experience Questionnaire: Development and Validation of a Measure for Assessing Recuperation and Unwinding from Work (2007) is a cross-sectional study validating the recovery-experience questionnaire. It found detachment most strongly related to well-being and reports nothing on next-day performance, which CC-6 Self-management had credited to it.
- Christina Maslach and Michael P. Leiter, Understanding the Burnout Experience: Recent Research and Its Implications for Psychiatry (2016) locates burnout's causes in six areas of the job and notes that interventions have favoured individual coping. It does not use the phrase "treating the person for the job's disease", which CC-4 Motivation, recognition, and sustainable pace had presented as the research's term; the unit now makes the point in its own words.
-

> Abi Noda, Margaret-Anne Storey, Nicole Forsgren and Michaela Greiler, DevEx: What Actually Drives Productivity (2023) proposes the developer-experience framework from its authors' earlier research and reports no new study. One author, not all of them, had led DORA. DE-5 Designing how work flows and the technique card now say so, and the model's evidence grade, on the unit and on its technique card, is corrected from research to practice.

A second pass the same day found the Sonnentag and Fritz claim still in the *Recovery and detachment* technique card, the glossary entry *Recovery* and a CC-6 Self-management self-check answer; all three now state the association with well-being only. It also restored v3.0's meaning in a few more places: the risk-list ordering in a DE-4 Managing scope, risk, and dependencies across workstreams self-check (uncertainty, *not* severity), a DE-5 Designing how work flows self-check (the flow, *not* the people), the CC-6 Self-management self-care tension (the test is managerial leverage), and firm statements the rewrite had softened in DE-1 Prioritisation under constraint, DE-4 Managing scope, risk, and dependencies across workstreams, DE-5 Designing how work flows, SV-1 Engineering strategy to SV-6 Organisational design and leading change, TJ-6 Security, privacy, and compliance as engineering responsibilities, PL-2 Feedback and difficult conversations and the Technical Judgment hub.

Erratum, 8 October: the framework, the index and *How Judgment Scales* said all three certifications "assess the same twenty-nine competencies". They draw on the same twenty-nine; the EMA-I outline gives PL-6 Leading through leaders and SV-6 Organisational design and leading change no questions. The framework itself is unchanged.

One version line

Before this release the framework was v1.0 (June 2026) and the Body of Knowledge was v2.0 (July 2026), each numbered on its own. They are one standard: the framework says what is examined, the Body of Knowledge teaches it, and the Exam Content Outlines say how each examination samples it. From v3.0 all three carry the same version, and a major version revises them together. A minor version may revise one of them and carries its number only there, as v3.1 does for the Body of Knowledge. Framework v2.0 does not exist; the number was skipped so that a version means one release everywhere it appears.

The framework: twenty-five competencies become twenty-nine

Five domains, equal weights and the five domain URLs are unchanged.

ID	Change	What changed and why
PL-1	Expanded	Adds decision rights as an explicit object: who may decide what, and how that authority

		is made visible. v1.0 treated delegation as handing off work; v3.0 treats it as designing who decides.
PL-2	Expanded	Names the difficult conversation as its own skill. The delayed, avoided or badly run conversation is the failure the examinations most often test.
PL-3	Expanded	Adds assessing capability rather than tool access: interviews that measure what a candidate can do, not which tools (including AI assistants) they were allowed in the room.
PL-4	Kept	Unchanged in scope. Fair process is stated rather than implied.
PL-5	Expanded	Widened from the one-on-one to the whole growth path: promotion decisions, the senior individual-contributor track, and the choice between it and management.
PL-6	**New**	Built from the v2.0 Professional-altitude material of PL-1, PL-4 and PL-5, which described leading managers as a variant of leading engineers. It is a different job with its own failure modes, and now has its own competency.
DE-1	Kept	Unchanged.
DE-2	Sharpened	Re-stated around uncertainty. The competency is not producing accurate estimates; it

CHANGELOG

		is making commitments that are honest about what is not known.
DE-3	Expanded	Reliability as an ongoing responsibility joins incident response: service-level objectives, the on-call system, and what an organisation learns from failure.
DE-4	Kept	Unchanged.
DE-5	Expanded	Widened from choosing a process to designing the flow of work: developer experience and AI-assisted workflows are part of what a manager designs.
SV-1	Expanded	Adds writing it down. A strategy that exists only in conversation cannot be argued with, delegated against or checked.
SV-2	Expanded	From communicating decisions outward to partnering: the relationship with product, design and executives is two-way.
SV-3	Kept	Unchanged.
SV-4	Expanded	Renamed from resource allocation and build-vs-buy to engineering economics: cost of delivery, budgets and headcount, vendors, and a third option beside build and buy.
SV-5	Expanded	Adds interpretation and the measurement of AI impact, where the pressure to show a number is strongest and the numbers are least settled.
SV-6	**New**	

		Built from the v2.0 Expert-altitude material of PL-1, which treated organisational structure as the executive form of delegation. Designing the organisation and leading it through change is a strategic competency and sits in Strategy & Vision.
TJ-1	Kept	Unchanged.
TJ-2	Kept	Unchanged.
TJ-3	Expanded	Adds standards for AI-generated code: what review must establish when the author did not write every line.
TJ-4	**Merged**	v1.0 TJ-4 (staying technically credible) and TJ-5 (judging when to intervene) are one judgment: you can only decide when to step in if you are close enough to see, and staying close is only worth it for the decisions it enables.
TJ-5	**New**	AI-assisted engineering: how an organisation adopts AI assistance, governs its use, builds the capability to use it well, and manages the risks. No v1.0 competency covered it. The ID TJ-5 is reused for a different competency; the v1.0 TJ-5 lives on inside v3.0 TJ-4.
TJ-6	**New**	Security, privacy and compliance as engineering responsibilities. v1.0 left them implicit inside quality and risk.
CC-1	Kept	Unchanged.

CC-2	Kept	Unchanged.
CC-3	Kept	Unchanged.
CC-4	Expanded	Adds distributed and hybrid teams, where recognition and pace are hardest to see.
CC-5	Expanded	From modelling values to exercising ethical judgment: the situations where stated values and the pressure of the moment point in different directions.
CC-6	**New**	Self-management: the manager's own time, energy, boundaries and resilience. No v1.0 competency addressed the manager as the object.

The Body of Knowledge: one text per competency

v2.0 wrote each competency as core concepts followed by three altitude bands (*At Associate*, *At Professional*, *At Expert*). v3.0 replaces the seventy-five bands with one unified text per competency, structured identically across all twenty-nine, and moves the treatment of altitude into a single chapter, *How Judgment Scales*. Nothing substantive from v2.0 is dropped; band material is de-duplicated and re-homed, most visibly the "developing managers" content into PL-6 and "designing ownership through structure" into SV-6.

Every unit section and every principle has a permanent anchor (`PL-1.P3`), so the standard can be cited at the level of a single claim. Every named model carries an evidence grade (`[research]`, `[practice]` or `[contested]`), and every source is a real, checkable work in one bibliography.

The v2.0 text remains readable, unchanged, in the archive.

Exam Content Outlines: new

CHANGELOG

Published for the first time, one per certification. Each states the examination's specification, the number of questions each competency carries, the proficiency indicators a certified holder meets, the scenario contexts questions draw on, the cognitive mix, what the examination does not assess, and the traceability of every competency to the Body of Knowledge. The EMA-I outline is complete; the EMP-II and EME-III outlines are complete drafts and say so until those examinations open.

Governance

A governance chapter states the separation of study material from examination development, the release cadence (annual major, quarterly minor), the public-comment process for major versions, the review statement, and the licence.

2026-10-05, within v3.0: the Exam Content Outlines' proficiency indicators are released under CC BY 4.0, joining the competency list; the rest of the standard stays CC BY-NC-ND 4.0. A How to Cite chapter is added.

Credentials

Credentials record the framework version they were examined against. Every credential issued before the v3.0 effective date carries `1.0`; the verification page and the verification API show it. Existing credentials are unaffected in every other respect.

Retired

- The document's former four-letter acronym, which collided with another body's long-standing one. The document is the *Engineering Management Academy Body of Knowledge*, short form *EMA Body of Knowledge*.
- The *Preparing for the Exams* chapter. Its guidance is carried by each Exam Content Outline's opening section and by the per-certification reading paths generated from the outlines.
- The altitude-band structure, as above.

Earlier releases

Body of Knowledge v2.0, July 2026. Five domain chapters rewritten with altitude bands; Techniques & Models, Preparing for the Exams and the glossary added; PDF edition.

Competency Framework v1.0, June 2026. Five domains, twenty-five competencies, equal weighting. The standard the EMA-I examination has been built against since launch.

Governance

How this standard is governed

The Engineering Management Academy publishes EMA Standard v3.0 as a certification standard. It consists of the Competency Framework, this Body of Knowledge and the Exam Content Outlines. This chapter states who maintains the standard, how it changes, how the public can comment on it, how study material is kept apart from examination development, and on what terms it may be used. These are commitments, and they are written down so that the Academy can be held to them.

Separation of study material and examination development

The Academy sells no training and endorses none. The Body of Knowledge is free, and it is the whole of the examined material: nothing a candidate is asked to know exists outside it. The credential's independence rests on that, and the independence depends on one internal rule, stated here so it can be checked.

The people who write the Body of Knowledge do not write examination items, and the people who write examination items do not see item performance while writing the Body of Knowledge. The units teach judgment and the items test it. Both are developed against the same framework and the same outlines, and never against each other. A unit's worked scenario is never drawn from the live item bank, and no item is ever written by copying a unit's scenario. The outlines are the only artifact the two share. Each outline states what a certified holder can do, and the units and the items are both written to that statement.

Where the Academy's size means one person holds both roles over time, the roles are held in sequence and the sequence is recorded. The Body of Knowledge is closed at a release before item development against it begins, and item development does not reopen it.

Release cadence

- **Major versions** change the framework's competencies, the outlines' weights or indicators, or the structure of the Body of Knowledge. They are released at most **once a year**, after a public comment period, with a changelog that explains every change and the reason for it.
- **Minor versions** carry corrections, added sources, clarified prose and new technique cards. They are released **quarterly**, and they do not change any competency, anchor, weight or indicator. Anchors are permanent: a minor version may add a principle at

the end of a unit's list, never renumber. A minor version carries its number only on the documents it revises. The framework, and so the version a credential records, changes only with a major version.

- **Errata** are factual corrections that cannot wait. They are issued as needed and recorded in the changelog.

Every credential records the version of the framework it was examined against, and each certification's Exam Content Outline states which version its examination currently uses. A candidate is always examined against a published, dated standard.

Public comment on major versions

Before releasing a major version, the Academy publishes the proposed changes (the framework diff, the outline diffs, and the units that change materially) for a comment period of not less than **thirty days**. Comments are received through the contact page and, where the commenter agrees, acknowledged by name in the release. The Academy publishes a summary of the comments it received and what it changed in response, and gives its reasons wherever it declined a comment. A comment is answered whether or not it is accepted.

Review

EMA Standard v3.0 was prepared and reviewed within the Academy before publication. Reviewers of a version are never named without their agreement, and none is named for v3.0. Review is not endorsement, and a reviewer holds no interest in the Academy.

Licence

The EMA Body of Knowledge and the Exam Content Outlines are published under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 licence, except for the two parts named in the next paragraph. You may copy, share, print and cite them, with attribution, for non-commercial purposes, without modification. To translate them, adapt them, or teach from them commercially, ask. The Academy answers such requests case by case.

Two parts of the standard are published under Creative Commons Attribution 4.0 instead. The first is the competency list of the EMA Competency Framework: the twenty-nine identifiers, titles and one-line descriptions. The second is the proficiency indicators of the three Exam Content Outlines, meaning each statement of what a certified Associate,

Professional or Expert can do, with its identifier and scope tag. Both may be copied, adapted and built on, commercially or not, with attribution, in job descriptions, career ladders, rubrics and tools. The rest of each outline (its specification, weights, scenario contexts and exclusions) stays under the licence above. A standard is adopted by being used, and the Academy wants the competency identifiers, and what each looks like in practice, to become a shared vocabulary.

The examination questions are not published and are licensed under neither.

Neither licence grants any right to use the names Engineering Management Academy, EMA-I, EMP-II, EME-III or EMX-AI, or the Academy's marks. A credential may be described by its name. A course or product may not be described as EMA-certified, EMA-approved or EMA-aligned.

Attribution

When citing the standard, cite the version and, where it matters, the anchor. *EMA Body of Knowledge v3.1, PL-1.P3* refers to one principle of one unit, and because anchors are permanent it will go on doing so. Suggested form: Engineering Management Academy, *Engineering Management Academy Body of Knowledge*, version 3.1, engineeringmanagement.academy/body-of-knowledge. The How to Cite chapter gives the APA, Chicago and BibTeX forms, and the attribution line for the openly licensed parts.

Corrections

Errors in the standard are corrected in the next minor version and recorded in the changelog. They include a source that does not say what the unit claims, a definition that contradicts a unit, and a broken anchor. Please report them through the contact page; the Academy would rather know.

How to Cite

Citing the standard

Cite the version and, where it matters, the anchor. Every competency, unit section, principle, model, tension and self-check question of EMA Standard v3.0 has a permanent identifier, and that identifier is the anchor in its URL. *EMA Body of Knowledge v3.1, PL-1.P3* is the third principle of the delegation unit, at the address ending `#PL-1.P3`. The same anchor named the same principle in v3.0, and it will keep doing so in every later minor version, because anchors are never renumbered except in a major version (release cadence). Anyone can check a citation that names the version and the anchor, and that is why a standard is cited this way.

The forms below follow APA 7, Chicago (bibliography) and BibTeX. The Engineering Management Academy is both author and publisher, and no individual is named. Every unit page has the APA form for that unit in its *Cite this unit* box, ready to copy.

The whole standard

APA:

> Engineering Management Academy. (2026). *Engineering Management Academy Body of Knowledge* (Version 3.1). https://www.engineeringmanagement.academy/body-of-knowledge

Chicago:

> Engineering Management Academy. *Engineering Management Academy Body of Knowledge*. Version 3.1. Engineering Management Academy, 2026. https://www.engineeringmanagement.academy/body-of-knowledge.

BibTeX:

```
@book{ema-bok-v3-1,
author    = {{Engineering Management Academy}},
title     = {Engineering Management Academy Body of Knowledge},
edition   = {Version 3.1},
publisher = {Engineering Management Academy},
year      = {2026},
url       =
```

```
{https://www.engineeringmanagement.academy/body-of-knowledge}
}
```

A knowledge unit

Cite the unit by its competency identifier and title, as part of the Body of Knowledge. The example below is PL-1 Delegation, ownership, and decision rights, and the same pattern holds for all twenty-nine.

APA:

> Engineering Management Academy. (2026). PL-1 Delegation, ownership, and decision rights. In *Engineering Management Academy Body of Knowledge* (Version 3.1). https://www.engineeringmanagement.academy/body-of-knowledge/people-leadership/delegation

Chicago:

> Engineering Management Academy. "PL-1 Delegation, ownership, and decision rights." In *Engineering Management Academy Body of Knowledge*, version 3.1. Engineering Management Academy, 2026. https://www.engineeringmanagement.academy/body-of-knowledge/people-leadership/delegation-.

BibTeX:

```
@incollection{ema-bok-v3-1-pl-1,
author    = {{Engineering Management Academy}},
title     = {PL-1 Delegation, ownership, and decision rights},
booktitle = {Engineering Management Academy Body of Knowledge},
edition   = {Version 3.1},
publisher = {Engineering Management Academy},
year      = {2026},
url       =
{https://www.engineeringmanagement.academy/body-of-knowledge/people-leadership/del
}
```

A single principle, section or question

The anchor is a locator inside the unit and goes where a page number would. In running text the short form is enough: *EMA Body of Knowledge v3.1, PL-1.P3*, or in APA's author–date style (Engineering Management Academy, 2026, PL-1.P3). In a reference list, cite the unit with the anchor as the locator and in the URL:

APA:

> Engineering Management Academy. (2026). PL-1 Delegation, ownership, and decision rights. In *Engineering Management Academy Body of Knowledge* (Version 3.1, PL-1.P3). https://www.engineeringmanagement.academy/body-of-knowledge/people-leadership/delegation#PL

Chicago:

> Engineering Management Academy. "PL-1 Delegation, ownership, and decision rights." In *Engineering Management Academy Body of Knowledge*, version 3.1, PL-1.P3. Engineering Management Academy, 2026. https://www.engineeringmanagement.academy/body-of-knowledge/people-leadership/delegation#PL .

BibTeX:

```
@incollection{ema-bok-v3-1-pl-1-p3,
author    = {{Engineering Management Academy}},
title     = {PL-1 Delegation, ownership, and decision rights},
booktitle = {Engineering Management Academy Body of Knowledge},
edition   = {Version 3.1},
publisher = {Engineering Management Academy},
year      = {2026},
note      = {PL-1.P3},
url       =
{https://www.engineeringmanagement.academy/body-of-knowledge/people-leadership/del
}
```

The anchor forms are `S1`–`S10` for a unit's sections, `P` for a principle, `M` for a model, `T` for a tension, `Q` for a self-check question, and `I` for a proficiency indicator on an Exam Content Outline. On the framework page a single competency is anchored by its identifier alone, as in `/framework#PL-1`.

The Competency Framework

APA:

> Engineering Management Academy. (2026). *EMA Competency Framework* (Version 3.0). https://www.engineeringmanagement.academy/framework

HOW TO CITE

Chicago:

> Engineering Management Academy. *EMA Competency Framework-*. Version 3.0. Engineering Management Academy, 2026. https://www.engineeringmanagement.academy/framework.

BibTeX:

```
@misc{ema-framework-v3-0,
author    = {{Engineering Management Academy}},
title     = {EMA Competency Framework},
edition   = {Version 3.0},
publisher = {Engineering Management Academy},
year      = {2026},
url       = {https://www.engineeringmanagement.academy/framework}
}
```

Which licence covers which part

Part of EMA Standard v3.0	Licence
The competency list of the Competency Framework: the twenty-nine identifiers, titles, domains and one-line descriptions	CC BY 4.0
The proficiency indicators of the three Exam Content Outlines: each "A certified ... can ..." statement, with its identifier and scope tag	CC BY 4.0
The Body of Knowledge: the units, the chapters, the techniques, the glossary and the sources	CC BY-NC-ND 4.0
The prose of the framework page, and the rest of each Exam Content Outline: its specification, weights, scenario contexts, cognitive mix and exclusions	CC BY-NC-ND 4.0
The examination questions	Not published, and not licensed
The names Engineering Management Academy, EMA-I, EMP-II, EME-III and EMX-AI, and the Academy's marks	Not licensed by either licence

HOW TO CITE

The two CC BY parts may be copied, adapted, translated and built on, commercially or not, as long as they are attributed. Everything under CC BY-NC-ND may be copied, shared, printed and cited, with attribution, for non-commercial purposes and without modification. To do more, ask through the contact page. The governance chapter states the terms in full.

Attributing the openly licensed parts

When you use the competency list or the proficiency indicators unchanged, in a career ladder, a rubric, a job description or a tool, put this line beside them:

> Competency list and proficiency indicators from the EMA Competency Framework v3.0 and its Exam Content Outlines, by the Engineering Management Academy (-https://www.engineeringmanagement.academy/framework), licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

If you have changed them (reworded an indicator, merged two competencies, added your own), say so, because the licence requires it:

> Adapted from the competency list and proficiency indicators of the EMA Competency Framework v3.0 and its Exam Content Outlines, by the Engineering Management Academy (https://www.engineeringmanagement.academy/framework), licensed under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/). Changes were made.

Both parts are also published as data, generated from the same source as the pages, with the licence and the attribution line inside each file: competencies.json and competencies.csv. Each competency in them links to its row on the framework page.

Attribution is not endorsement. Using the competency list does not make a course, a ladder or a product EMA-certified, EMA-approved or EMA-aligned, and none may be described that way.