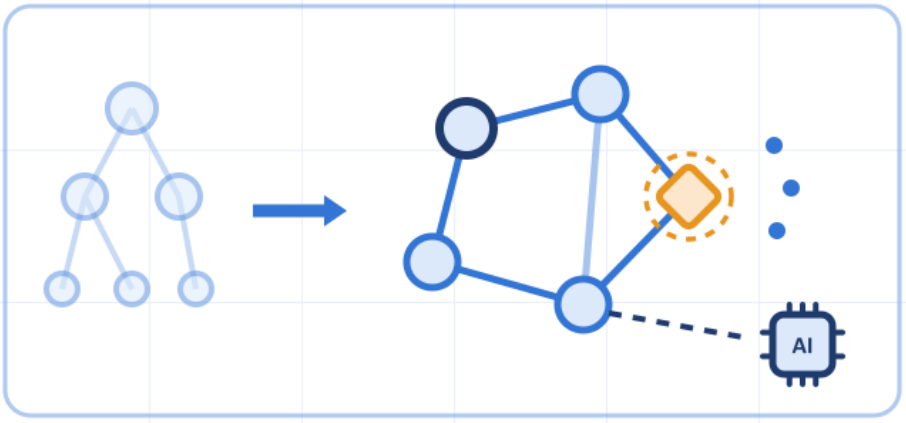
Scrum Manager
Uncovering better ways

Guide

Agile Leadership – Decision Architecture

Decision-making in agile teams

In-depth development of the State of the art topics



Updated June 2026

Scrum Manager® · Skill Arena

About this document

This document is an in-depth development of the topics in the reference map (State of the art) for designing how decisions are made in agile teams: where they are made, with what information, at what speed, with what safety net and what part AI plays at each point of the flow, which is available at: Skill Arena.

Use it as reference material to keep your practice aligned with, and checked against, current professional knowledge.

It is complemented by the training and assessment platform in Skill Arena. In the Agile Leadership – Decision Architecture Decision-making in agile teams area you can take training tests to check and improve your level of knowledge and, if you wish, you can also obtain a diploma that provides curricular accreditation of professional competence and currency in this area.

State of knowledge

Knowledge in this area evolves at two speeds. Its foundations —intent-based leadership, reversibility, decision rights, deciding under uncertainty— have been settled for decades and change slowly. The human–AI decision layer, by contrast, evolves extremely fast: frameworks, adoption data and regulation move month by month, and account for most of the revisions of the reference map. Throughout the document, the labels (ESTABLISHED, CONSOLIDATING, EMERGING) help identify the maturity of each concept:

ESTABLISHED settled consensus; knowledge taken as required.

CONSOLIDATING gaining adoption fast; not yet universal but already relevant.

EMERGING recent frontier; high relevance and high volatility.

Contents

Chapters of the guide, in the order of the State of the art.

BLOCK 1 · FROM THE LEADER WHO DECIDES TO THE LEADER WHO DESIGNS

Chapter 1. Decision architecture as the object of agile leadership ·

CONSOLIDATING

Traditional leadership consists of making decisions; agile leadership consists of designing the system that makes them.

Introduction

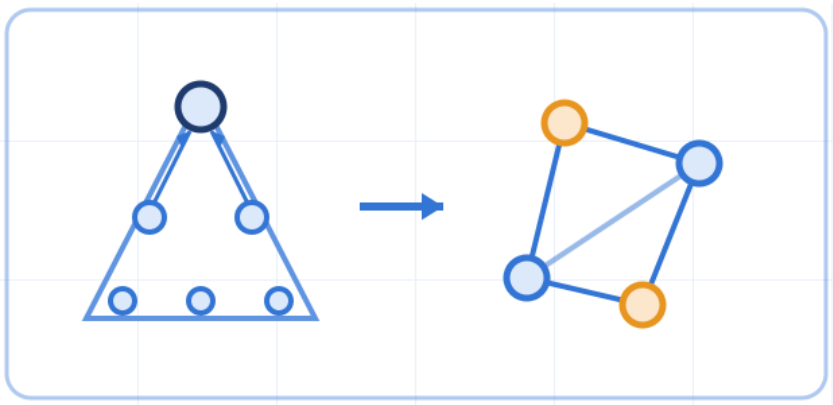
When you ask what distinguishes a good agile leader, the usual answers orbit around inspiration: vision, purpose, servant leadership, culture. All of that is true. And all of it is, in practice, inoperative: it does not say what to do on Monday morning when a delivery slips because nobody knew whether they could decide without asking. This chapter proposes a different and far more concrete object of work for leadership: the team's decision architecture.

The thesis running through this whole topic is simple to state: the difference between teams that work and teams that get stuck lies not in the charisma of whoever leads them, but in how their decisions are designed and distributed. Who decides what, with what information, at what speed, with what safety net and —increasingly— with what part played by AI at each point of the flow.

1.1 The decision as a flow of the system, not an attribute of the boss

The inherited mental model treats a decision as an attribute of position: whoever is in charge decides. Under that premise, information flows upwards —it is collected, summarised, presented— and the decision flows downwards in the form of an instruction. Every hop of that journey has two costs that appear in no budget: latency (the time the decision spends in transit and in queues) and loss of context (each summary strips out nuances that whoever decides can no longer recover).

Seen as a system, the problem stops being about people and becomes a matter of design. A team where every relevant decision travels two levels up and back is not a team that is slow for lack of talent: it is a system with the decision badly placed with respect to the information. The agility of an organisation is, to a large extent, the sum of the distances its decisions travel.



From the apex that concentrates decisions to the network that distributes them by design.

Core idea. The question of agile leadership is not "what do I decide?" but "where should this decision be made, with what information, at what speed and with what safety net?". Whoever

answers that question well multiplies their capacity: they no longer decide once, they improve every future decision of the system.

1.2 The architecture always exists, even if nobody designed it

The uncomfortable truth: agile teams almost never have an explicit decision map. Everybody knows —more or less, and each in their own way— who can decide what, until a specific case proves they do not. The decision architecture always exists; what happens is that it has grown on its own, by sedimentation of habits, personalities and historical accidents. Like technical debt: invisible while untouched, extremely expensive when it has to be changed.

This implicit architecture has recognisable symptoms that chapter 5 develops in detail: answers along the lines of "it depends" or "ask so-and-so", decisions that bounce between people with no clear owner, and the awkward silence when somebody asks out loud who can, for example, drop a story from the backlog. The leader-designer's first act is not to change anything: it is to make visible what is already there.

1.3 Decision velocity: the competence of 2026

The shift of focus is not merely conceptual: talent trend reports reflect it. DHR Global's report for 2026 places agile leadership operating at "decision velocity" as the top competence of the year: leaders able to decide with incomplete information, correct course quickly and maintain clarity of communication amid constant change. The report also stresses that decision speed must be built as an organisational capability, not as the heroic trait of a particular leader — which is exactly the difference between making decisions and designing how they are made.

Uncertainty is the default state, and delaying decisions is in itself a risk. But accelerating without design produces chaos, not speed. The following chapters build, piece by piece, the system that allows fast decisions without bad ones: authority placed next to the information (block 2), reversibility as a criterion (block 3), decision rights and records (block 4) and heuristics for uncertainty (block 5), before adding the AI layer (blocks 6 to 8).

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the thesis of decision architecture and contrast it with the model of leadership based on making decisions.
- Identify, in a real workflow, the latency and loss-of-context costs introduced by each hierarchical hop of a decision.
- Recognise the symptoms of an implicit decision architecture and argue why making it visible is the first step of its redesign.
- Relate "decision velocity" to the design of the decision system, distinguishing it from mere individual speed.

Common mistakes and antipatterns

Confusing leadership with decision-making prominence. Measuring your own value by the number of decisions that pass through you. It is the inverse of the correct metric: a well-designed system needs its designer less and less in the day-to-day.

Staying at the level of inspirational discourse. Vision and purpose without architecture produce motivated teams that get stuck at the same bottlenecks. Inspiration does not replace design; it accompanies it.

Treating slowness as a problem of attitude. Diagnosing "lack of initiative" where there is an architecture that punishes deciding without permission. People adapt rationally to the system they work in.

Redesigning without mapping first. Handing out authority over an architecture nobody has made explicit reorganises the confusion instead of removing it.

Further reading

· DHR Global — Talent Trends Outlook 2026

· Agile leadership: don't make the decisions, design how they are made — Scrum Manager

BLOCK 1 · FROM THE LEADER WHO DECIDES TO THE LEADER WHO DESIGNS

Chapter 2. Decision Intelligence: the emerging discipline · EMERGING

A discipline with two senses that are worth keeping apart: the professional competence of deciding better, and the category of technology platforms that Gartner established as a market in 2026.

Introduction

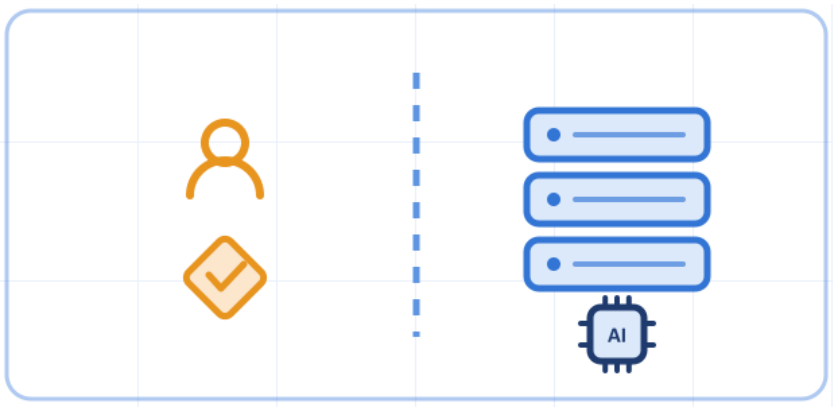
The term "Decision Intelligence" was born in the world of data. Cassie Kozyrkov, who served as Chief Decision Scientist at Google before founding their own firm, defined it as the discipline of turning information into better actions at any scale: a combination of data science with the social and management sciences, aimed not at producing analysis but at producing decisions. The starting proposition still holds: most organisations accumulate data; very few have professionalised the act of deciding with it.

Since January 2026 the term has also had a second life: Gartner published its inaugural Magic Quadrant for Decision Intelligence Platforms, consecrating a software market category. For an agile leader, handling both senses is not terminological pedantry: it is knowing what is being talked about when the word reaches —and it will reach— their organisation.

2.1 The competence sense: deciding as a professional discipline

In its original sense, Decision Intelligence is a competence of people and teams: treating a decision as a process that can be designed, measured and improved, instead of as a spontaneous act of intuition or authority. It includes knowing how to frame the decision before looking for data (what will be decided according to which criteria, set before seeing the information, in order to neutralise confirmation bias), telling decisions that deserve process from those that do not, and evaluating the quality of the process independently of the outcome.

This is the sense this topic develops, applied to the agile context: the competence does not consist of deciding more, but of better designing the system by which the team decides. Every block of this guide —from placed authority to delegating to agents— is a technique of that discipline.



Two senses of the same term: the professional competence of deciding (this topic) and the technology platform.

2.2 The platform sense: the market category

Decision Intelligence platforms are software for modelling, executing, monitoring and governing business decisions at enterprise scale, combining data, analytics, rules, AI and human judgement. Gartner's inaugural Magic Quadrant (January 2026) formalised the category and describes it as a late emerging market, in transition from niche adoption to strategic enabler. The underlying promise: organisations are not short of information, but of control over how information turns into action.

For the agile leader, the arrival of these platforms poses exactly the type of decision this topic teaches you to design: which of the organisation's decisions are delegated to a platform, which are kept in assisted mode and which remain human. It is the decision map of chapter 5 and the delegation logic of block 8, applied to a software vendor instead of to an agent on the team. Whoever has no judgement of their own will receive it, by default, from the vendor.

An essential distinction. When somebody says "Decision Intelligence", ask which of the two senses they mean. The competence belongs to people and is trained; the platform is a product and is bought. Buying the second without developing the first is equivalent to delegating decisions without the two pillars of chapter 4: it is not decentralising judgement, it is abandoning it.

2.3 Why an agile leader should know both

The consolidated certifications in the agile leadership space focus on mindset, leadership stances, culture and transformation. None deals systematically with leadership as the design of decision architectures, nor with the AI layer over judgement. That is this skill's specific contribution to the professional profile: an operative competence —designing decision systems— at the very moment the market starts selling decision systems.

The order matters: competence first, tool afterwards. A team that has made its decision map explicit, that classifies by reversibility and that knows how to evaluate decision processes is in a position to make the most of a DI platform; a team without that groundwork will only automate its confusion, faster.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Tell the two senses of Decision Intelligence apart —professional competence and technology platform— and explain the origin and trajectory of each.
- Explain what the discipline of Decision Intelligence adds over traditional data analysis: the focus on the decision and on the quality of the decision-making process.
- Argue which conditions of competence an organisation should have before delegating decisions to a DI platform.
- Place this skill on the map of agile leadership certifications, identifying its differentiator: leadership as the design of decision architectures with an AI layer.

Common mistakes and antipatterns

Confusing the two senses. Arguing about the competence with product arguments, or the other way round. It leads to buying software expecting it to solve a problem of organisational design, or to dismissing useful tools out of generic distrust.

Platform first, judgement later. Adopting a DI platform without a prior decision map. The platform will execute an architecture: if nobody has designed it, it will execute the implicit one, with its defects automated.

Reducing Decision Intelligence to analytics. Treating the discipline as "more dashboards". The dashboard informs; the discipline designs who decides with it, when and against which criteria set in advance.

Further reading

· Kozyrkov — Introduction to Decision Intelligence

· Gartner — Magic Quadrant for Decision Intelligence Platforms (2026)

BLOCK 2 · MOVING AUTHORITY TO THE INFORMATION

Chapter 3. Intent-based leadership · ESTABLISHED

Instead of moving information towards authority, move authority to where the information is.

Introduction

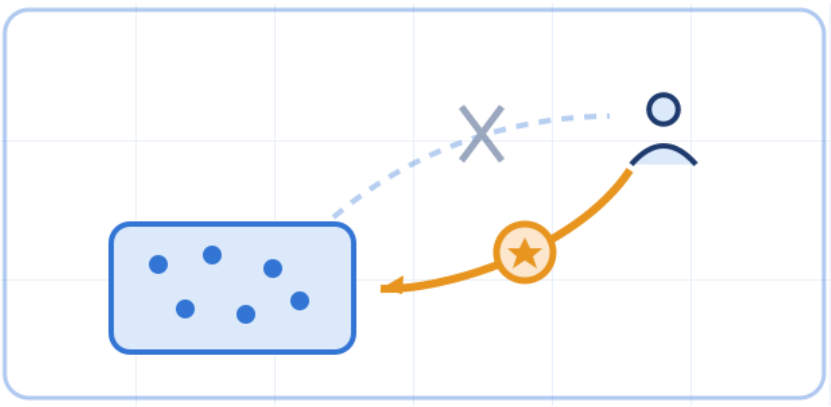
In 1999, David Marquet took command of the nuclear submarine USS Santa Fe, a different boat from the one he had spent a year preparing to lead. Trained in the "whoever is in charge knows everything and orders everything" model, he found himself giving orders he could not justify to a crew trained to obey them without question. Out of that crisis came the experiment he recounts in Turn the Ship Around!: stop giving orders and get the crew —who did have the information— to take on the authority to decide.

The principle he distilled is the basis of this block: instead of moving information towards authority, move authority to where the information is. The Santa Fe went from the worst results in the fleet to the best, and the model, named intent-based leadership, became one of the most widely adopted leadership references of recent decades.

3.1 The principle: authority where the information is

Every decision needs two ingredients: information about the situation and authority to act. The traditional hierarchical model keeps them apart —information at the bottom, authority at the top— and devotes enormous energy to transporting the former towards the latter: reports, status meetings, escalations. The transport degrades the information (chapter 1) and delays the action.

Marquet's alternative reverses the flow: authority comes down to where the information already is. Whoever operates the sonar decides about the sonar; whoever talks to the customer decides on the nuance of the conversation. This is also the real operating basis of the "self-organising teams" of the agile frameworks: self-organisation is not the absence of authority, but authority placed next to the information, with the safeguards the next chapter details.



Authority travels towards the information, instead of pushing information up towards authority.

3.2 The mechanics of "I intend to..."

The practical genius of the model lies in its linguistic mechanics. Instead of asking permission ("may I submerge the boat?") or waiting for orders, the operator states their intent: "Captain, I intend to

submerge the boat". The statement obliges whoever makes it to have thought the decision through in full —what they are going to do, why, what they have checked— and allows whoever hears it to verify the reasoning without taking the decision away. The usual answer is, simply, "very well".

"I intend to..." works as a mechanism for the progressive transfer of authority: at first, intentions are voiced and confirmed; with time and demonstrated trust, whole categories of decision stop needing confirmation. It is delegation with a safety net: control is not let go all at once, it is transferred in an observable and reversible way. Carried over to agile work, it replaces "what do I do with this?" with "this is what I am going to do, unless you see something I do not".

Change of language, change of ownership. Moving from "I request permission to..." to "I intend to..." looks cosmetic and is not: it transfers ownership of the reasoning. Whoever asks permission outsources the thinking; whoever states an intent has already done it. A team's language is a reliable indicator of where its authority really lives.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the principle of moving authority to the information and its origin in the Santa Fe case.
- Apply the mechanics of "I intend to..." to transfer authority progressively and observably in a team.
- Reinterpret the self-organisation of the agile frameworks as authority placed next to the information, with gradual and verifiable transfer.
- Detect, in a team's everyday language (requests for permission as against statements of intent), where authority in fact resides.

Common mistakes and antipatterns

Delegating all at once and without a net. Going from total control to total autonomy in a single announcement. The transfer of authority is progressive precisely so that trust and competence grow at the same pace as autonomy.

Taking authority back at the first mistake. A failure after delegating usually prompts immediate recentralisation, which teaches the team that the autonomy was decorative. The mistake calls for adjusting the transfer, not reversing it.

Intent without verification. Accepting every "I intend to..." without listening to the reasoning turns the mechanism into a ritual. The verification phase is where the leader contributes context and where the transfer is calibrated.

Confusing self-organisation with the absence of design. Proclaiming the team self-organising without defining what authority has been moved, over which decisions and within which limits. That is not self-organisation: it is the implicit architecture of chapter 1 under another name.

Further reading

· Marquet — Turn the Ship Around!

· Intent-Based Leadership resources

BLOCK 2 · MOVING AUTHORITY TO THE INFORMATION

Chapter 4. The two pillars: competence and clarity of intent · ESTABLISHED

Decentralising is not handing out decisions and wishing everyone luck: without competence and without clarity of intent, it is not decentralising, it is abandoning.

Introduction

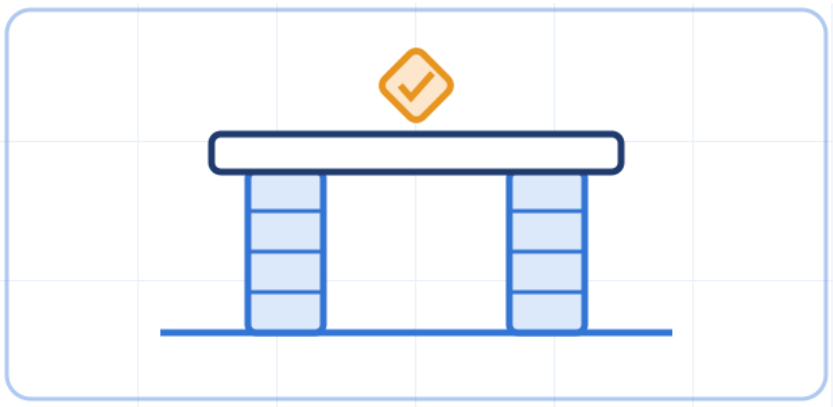
Enthusiasm for decentralisation has a characteristic failure mode: handing out decisions and wishing everyone luck. Marquet himself is emphatic on this point: giving control on its own is a recipe for disaster. Authority can only move to where the information is if whoever receives it can sustain it, and sustaining it takes exactly two things: knowing how, and knowing what for.

This chapter develops those two pillars and the format that makes them operative —the commander's intent— and shows why this classic discipline of delegation has become, with the arrival of AI, a dual-use competence.

4.1 Pillar 1: technical competence

Whoever decides has to know how. It seems obvious and is constantly breached: the architecture decision is delegated to somebody who has never operated the system in production, or prioritisation to somebody who does not know the market. Competence is not a moral requirement but a functional one: without it, the person will decide late (out of insecurity), badly (out of ignorance) or by reverse delegation (returning the decision disguised as a question).

The design consequence is that delegating implies investing: training, support, gradual exposure to decisions of greater weight. In Marquet's model, every transfer of authority is accompanied by a deliberate reinforcement of competence. Chapter 23 picks up this idea from the other end: what happens to decision-making competence when AI absorbs the decisions it used to be trained on.



A delegated decision rests on two pillars: technical competence and clarity of intent. If one is missing, it falls.

4.2 Pillar 2: clarity of intent and the commander's intent format

The second pillar answers the "what for". A competent person without clarity of intent optimises blindly: they decide well towards the wrong objective. The military tradition solved this problem with a format that has aged very well, the commander's intent: a brief statement of what is sought, what constraints exist and what is non-negotiable. Its function is to let any subordinate decide well

in the face of the unforeseen without asking for new orders, because they know the purpose of the mission, not just its steps.

Carried over to agile work: replace instructions with intent. Instead of "implement these five validations", a commander's intent would say "no corrupt data is to reach billing (objective); without degrading response time below X (constraint); regulatory compliance is not negotiable (non-negotiable)". With that, the team can resolve the thousand cases the instruction did not foresee.

A discipline of delegation. Formulating a good commander's intent costs more than giving an instruction, and that is exactly its virtue: it forces whoever delegates to distil what they really want. If you cannot write down the objective, the constraints and the non-negotiables of what you are delegating, you do not yet know what you are delegating.

4.3 The same discipline AI demands: the connection with SDD

This discipline has found an unexpected second field of application: specifying work for an AI. A good specification in Spec-Driven Development is intent formalised for an agent —what is wanted and why, within which constraints, what is non-negotiable— exactly as commander's intent is intent formalised for a team. Whoever knows how to delegate to people with clear intent already has half the craft of directing agents; whoever delegates with vague instructions will fail equally with both.

The connection also works in reverse: practising the writing of specifications for AI trains the muscle of making intent explicit, and that training improves human delegation. The topic SDD in agile teams develops the full discipline on the AI side.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Formulate a complete commander's intent —objective, constraints, non-negotiables— for a real delegation.
- Diagnose whether a delegation meets the two pillars or is abandonment, and identify which pillar is missing.
- Design the reinforcement of competence that should accompany a specific transfer of authority.
- Explain the equivalence between commander's intent and a specification for an AI, and apply it in both directions.

Common mistakes and antipatterns

Decentralisation by proclamation. Announcing that "the team decides now" without building the two pillars. The predictable result —late or bad decisions— is then used as proof that "autonomy does not work".

Instructions disguised as intent. Drafting a supposed commander's intent that actually lists the steps to follow. If the document leaves no room for decision, it does not transfer authority: it micromanages in writing.

Intent without non-negotiables. Stating the objective while omitting constraints and red lines. The person delegated to will discover them by crossing them, and mutual trust will pick up the bill.

Confusing consultation with reverse delegation. Systematically accepting that the delegated decision comes back in the form of a question ("what would you do?"). Consulting for context is healthy; handing the decision back is not.

Further reading

· SDD in agile teams — Skill Arena

· Marquet — Turn the Ship Around!

BLOCK 2 · MOVING AUTHORITY TO THE INFORMATION

Chapter 5. The team's decision map · CONSOLIDATING

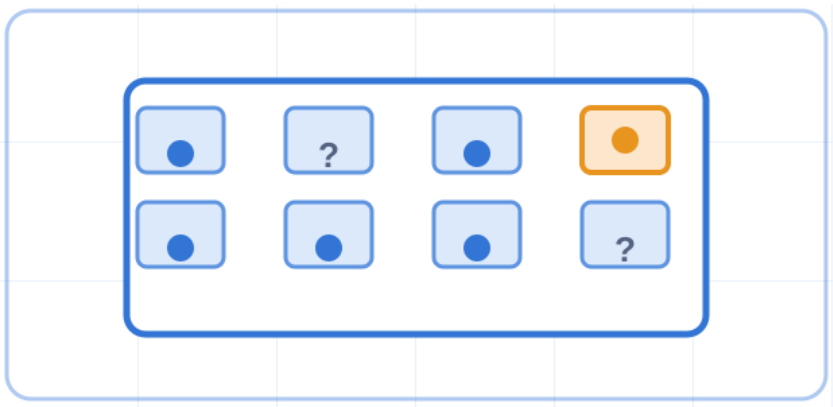
You cannot redesign an architecture that has not been made visible: the decision inventory is the starting practice.

Introduction

The previous chapters establish the principle (authority to the information) and its conditions (competence and intent). This chapter contributes the practice to start with tomorrow: the team's decision map. It is the decision-making equivalent of making the workflow visible: before optimising anything, see what is there.

5.1 The decision inventory

The practice consists of answering, as a team and in writing, three questions about recurrent decisions: which decisions are made over and over (changing a dependency, dropping a story, delaying a delivery, granting a customer an exception...), who in fact makes them today —not who should— and who has the best information to make them. The third column compared with the second is the diagnosis: every mismatch is avoidable latency and loss of context.



The inventory board: recurrent decisions, some with an explicit owner and others still with a question mark.

The natural format is a single-topic retrospective: a session devoted only to this, with the decisions of the last couple of months as raw material. Exhaustiveness is not needed; with the fifteen or twenty most frequent or most contentious decisions, the map already reveals the pattern. Calibration tools such as delegation poker (chapter 8) fit naturally as the second part of the session, to agree the level of delegation for each decision in the inventory.

5.2 Symptoms of an implicit architecture

The inventory usually runs into three symptomatic answers. The first is "it depends": the decision has no owner, it has case law; it is resolved differently depending on who is around that day. The second is "ask so-and-so": the architecture lives in one person's head, and that person is both a bottleneck and a single point of failure. The third is silence: nobody knows who can decide, so nobody decides, and the decision makes itself by omission — which is the worst way of making it.

A quick test. In the next team meeting, ask out loud: "who can decide that we delay the delivery by a week?". If the answer takes more than five seconds or begins with "it depends", the team has an implicit architecture. This is not a judgement about the people: it is the natural state of any architecture nobody has designed.

5.3 From the map to the redesign

The map is not the end but the beginning. With the inventory in front of you, the redesign applies the criteria of the rest of the guide: reversible decisions go down to the level closest to the information (chapter 6); recurrent or contentious ones get explicit rights with DACI or RAPID (chapter 8); each decision is matched with its mechanism —consent, advice process— according to its cost and its type (chapter 9); and significant ones leave a record (chapter 10).

The map is best treated as a living artefact: review it when the team or the context changes, and extend it when AI agents with the capacity to decide arrive. Chapter 21 picks up exactly this artefact to add the lanes of hybrid work: what the agent decides, what it proposes and what remains human.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Facilitate a single-topic decision inventory retrospective, obtaining the three columns: what is decided, who decides today and who has the best information.
- Diagnose the mismatches between who decides and who has the information, and prioritise which to correct first.
- Recognise the three symptoms of an implicit architecture ("it depends", "ask so-and-so", silence) and explain what each reveals.
- Connect the map with the instruments of redesign: reversibility, decision rights, mechanisms and records.

Common mistakes and antipatterns

Mapping what ought to be, not what is. Filling in the inventory with the official organisation chart instead of with reality. The value of the map lies in portraying the de facto architecture, with its shortcuts and its dead zones.

Seeking exhaustiveness. Trying to inventory every decision paralyses the practice. The fifteen or twenty recurrent or contentious ones are enough to start; the rest are added when they hurt.

A map without redesign. Holding the session, putting the result on the wall and changing nothing. A map that does not lead to moving at least one decision in the next sprint is perceived as bureaucracy and burns the practice.

Turning the map into a rulebook. Formalising even trivial decisions. The criterion of chapter 8 applies from day one: make explicit what is recurrent or contentious, do not bureaucratise what is trivial.

Further reading

· Management 3.0 — Delegation Poker

· Agile leadership: don't make the decisions, design how they are made — Scrum Manager Observatory

BLOCK 3 · REVERSIBILITY AS A DESIGN CRITERION

Chapter 6. Reversible and irreversible decisions · ESTABLISHED

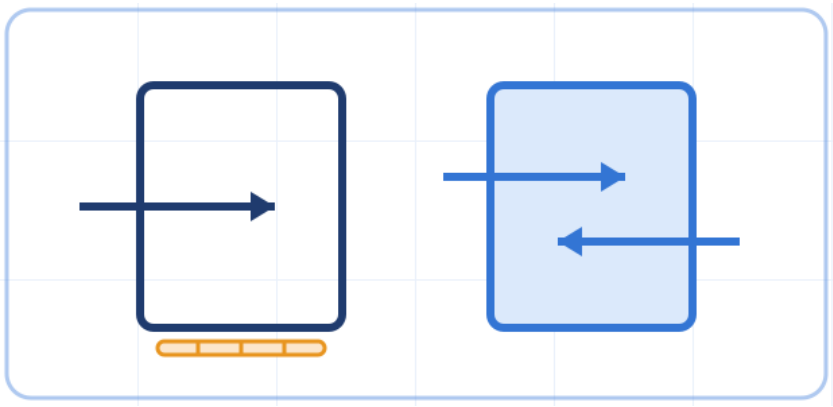
Classify by reversibility, not by importance: "important" is subjective; reversible or irreversible is operative.

Introduction

If block 2 answers where to place authority, this block contributes the criterion for grading it. Jeff Bezos formulated it in his letters to Amazon shareholders with an image that has caught on: there are decisions that are one-way doors —consequential and irreversible or nearly so; if you go through and do not like what you see, you cannot come back— and decisions that are two-way doors: if the decision turns out to be suboptimal, you reopen the door and go back. Most decisions, he warned, are of the second kind.

6.1 Two types of door, two decision processes

The classification is not taxonomy for its own sake: each type demands a different process. One-way doors should be decided methodically, slowly, with deliberation and consultation: the cost of being wrong is high and there is no way back. Two-way doors should be decided fast, by individuals or small groups with good judgement, without ceremony: the cost of being wrong is low and the cost of over-deliberating, high.



One-way doors —crossed slowly and with deliberation— and two-way doors, which allow you to go back.

The two possible mistakes are symmetrical and both abound. The first: treating every decision as irreversible —committees, approval chains, paralysis— for things that can be undone in an afternoon; Bezos pointed to this as the disease of organisations as they grow, with its sequel of slowness, unreflective risk aversion and lack of experimentation. The second: treating irreversible decisions with the lightness of reversible ones; its monuments are the monolith impossible to migrate and the contract with no exit clause.

6.2 Why reversibility and not importance

The usual objection is to classify by importance. The problem is that "important" is subjective and, in practice, tends to mean "it makes me nervous": each person draws the boundary according to their anxiety, not according to the nature of the decision. Reversible or irreversible, by contrast, is a

question with a verifiable answer: can we undo it? at what cost? how long would it take? Two reasonable people arrive at the same classification; with "importance" that rarely happens.

Design rule. Reversible decisions go down to the level closest to the information, with explicit permission to be wrong. Irreversible ones go up only as far as needed —not all the way to the top: as far as needed—, are deliberately slowed down and are documented. This single rule, applied to the decision map of chapter 5, resolves most of the mismatches the inventory reveals.

6.3 Reversibility as the criterion that articulates the system

Classification by reversibility is the criterion that articulates the rest of the guide, and that is why this block comes before the other instruments. It decides which decision mechanism to use (chapter 9: light consent for what is reversible, deliberation for what is irreversible), when to apply the 70% heuristic (chapter 12: almost always, for what is reversible), where to reintroduce friction when AI accelerates the flow (chapter 20) and what can be delegated to an agent (chapter 21: always start with what is reversible).

What you should be able to do

Having mastered this chapter, a professional is able to:

- Classify a set of real decisions by reversibility, justifying each classification with the cost and time of undoing.
- Assign each type of door its appropriate process: speed without ceremony for reversible ones, deliberation and documentation for irreversible ones.
- Identify in an organisation the two symmetrical mistakes: bureaucratising what is reversible and treating what is irreversible frivolously.
- Argue why reversibility is an operative criterion superior to perceived "importance".

Common mistakes and antipatterns

One process for everything. Putting every decision through the same approval circuit, whatever its reversibility. It is the origin of the organisational slowness that is later treated with "more agility" without touching the cause.

Classifying by nervousness. Treating as irreversible what is merely uncomfortable. The right question is not "does this worry me?" but "can it be undone, at what cost and in how long?".

Irreversible by oversight. Signing off one-way doors without recognising them as such: dependencies that are hard to replace, data whose structure can no longer be changed, public commitments. Chapter 10 (records) is the vaccine: documenting forces you to look.

Permission to be wrong in words only. Pushing reversible decisions down but penalising the first mistake. If being wrong about something reversible carries a punishment, the team will push decisions back up, and rightly so.

Further reading

· Bezos — 2015 letter to shareholders

· Bezos — 2016 letter to shareholders

BLOCK 3 · REVERSIBILITY AS A DESIGN CRITERION

Chapter 7. Turning irreversible decisions into reversible ones · ESTABLISHED

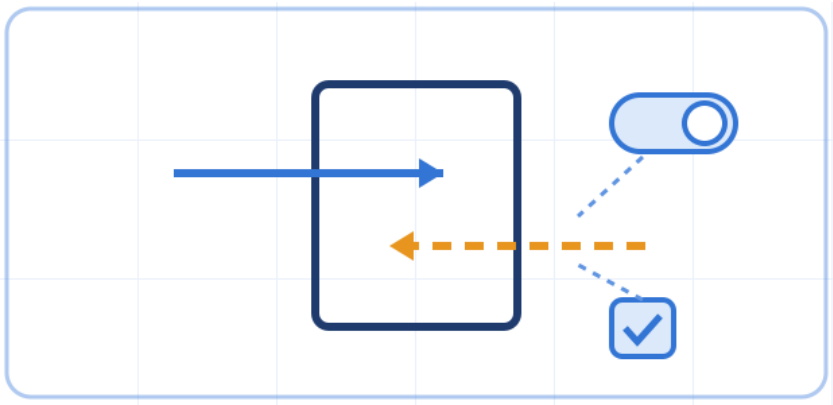
The boundary between reversible and irreversible is not fixed: every time undoing becomes cheaper, the zone in which the team decides without asking permission widens.

Introduction

The previous chapter treats reversibility as a given property of each decision. This one treats it as what it really is: a design variable. The boundary between one-way and two-way doors is not fixed by nature; it is fixed by the cost of undoing, and that cost can be worked on. A good part of the best engineering of recent decades consists, seen from here, of moving decisions from the expensive category to the cheap one.

7.1 Technical infrastructure as decision architecture

Automated tests, feature flags, incremental deployments, decoupled architectures, infrastructure as code: the usual catalogue of good technical practice shares an economic function that is rarely stated: making the cost of undoing cheaper. With a solid test suite, changing a library stops being a gamble; with feature flags, switching a feature on stops being a commitment; with incremental deployments, a production error stops being a crisis.



A way back is fitted to the one-way door: tests, switches and incremental deployments make undoing cheaper.

The organisational consequence is the key conclusion of the chapter: technical infrastructure is also decision architecture. Every decision that moves from irreversible to reversible can, by the rule of chapter 6, go down to the level closest to the information without asking permission. That is why a team with good engineering does not only deliver better: it decides faster, because its zone of autonomous decision is wider. And that is why cutting back on technical quality is, silently, recentralising decisions.

Business case. When it is hard to justify investing in tests or in decoupling an architecture, translate it into decisions: "this turns N decisions from irreversible into reversible, which lets the team make them without escalating". It is the argument a management committee understands, because it speaks of decision speed, not of technical virtue.

7.2 The principle outside software

The pattern travels well beyond code. An exit clause turns a single-door contract into a two-way door with a known toll. A pilot turns an irreversible organisational rollout into a bounded experiment. Phased commitments —investing in tranches with decision points— make reversible in parts what would not be in one piece. A temporary contract or a prior collaboration makes undoing a hire cheaper.

The design question is always the same: what would we have to add to this decision to be able to undo it at a reasonable cost? Sometimes the answer is expensive and not worth it —there are legitimate irreversible decisions— but with surprising frequency reversibility costs a clause, a phase or a test that nobody had asked for because nobody had asked the question.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Propose, for a specific irreversible decision, at least one intervention that makes it reversible or reduces the cost of undoing it.
- Explain common technical practices (tests, feature flags, incremental deployments) in terms of their effect on decision architecture.
- Carry the principle beyond software: exit clauses, pilots and phased commitments.
- Argue the investment in technical quality to the business in terms of decision speed and autonomy.

Common mistakes and antipatterns

Paper reversibility. Relying on ways back that have never been rehearsed. A rollback that has never been executed is a hypothesis, not a two-way door; reversibility is tested, like backups.

Making undoing cheaper without widening autonomy. Investing in tests and flags and keeping the same approval circuit. If the zone of autonomous decision does not widen, the investment yields half: reversibility has been paid for without collecting the speed.

Reversibilising everything. Pursuing the reversibility of decisions where its cost exceeds the benefit. There are legitimate one-way doors; the aim is to choose them consciously, not to eliminate them.

Ignoring organisational reversibility. Applying the principle to code only. Contracts, public commitments and team structures are the most expensive irreversible decisions of most organisations.

Further reading

· One-way door decisions in software development

· Bezos — 2015 letter to shareholders

BLOCK 4 · DECISION-RIGHTS FRAMEWORKS AND RECORD-KEEPING

Chapter 8. Frameworks for assigning decision rights · ESTABLISHED

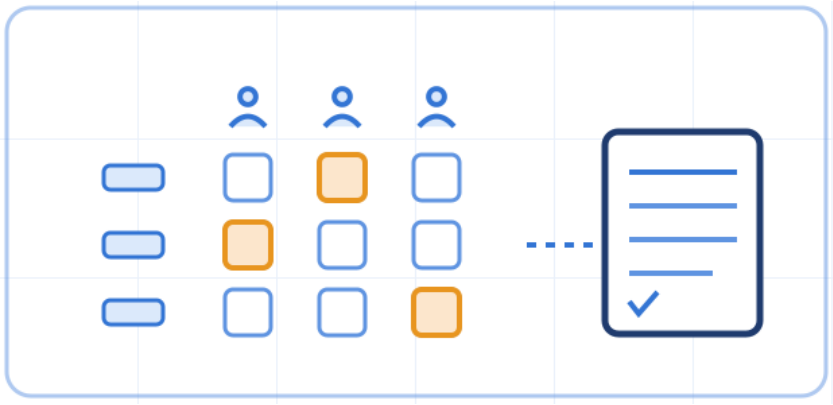
Making explicit who plays which part in each decision: formalise the recurrent or contentious ones, do not bureaucratise the trivial ones.

Introduction

The decision map of chapter 5 reveals the mismatches; this chapter contributes the vocabulary to correct them. Decision-rights frameworks make explicit, decision by decision, who drives the process, who contributes, who approves and who simply needs to be told. Their value lies not in sophistication —they are tables with roles— but in removing the ambiguity that turns simple decisions into endless negotiations.

8.1 DACI: cross-cutting day-to-day decisions

DACI, popularised by Atlassian's Team Playbook, assigns four roles: Driver (a single person who drives the process: gathers information, convenes, sets deadlines and gets the decision made on time), Approver (whoever approves: ideally one), Contributors (those who contribute knowledge or data, without a vote) and Informed (those who are affected and must know the result without taking part in it). Its natural ground is cross-cutting day-to-day decisions: those that cross several teams or functions and, without explicit roles, drag on forever because everybody has an opinion and nobody drives.



Explicit decision rights per decision and per role, with their record: what was decided, what was ruled out and what was known.

8.2 RAPID: complex multi-level governance

RAPID, developed by Bain, breaks the decision down into five parts: Recommend (whoever produces the proposal with its analysis), Agree (those with a formal right of veto, typically legal or compliance), Perform (those who will execute what is decided), Input (those who are consulted, without a veto) and Decide (a single person with the "D": the final authority). Its characteristic contribution is separating the veto (Agree) from mere consultation (Input), and requiring a single decision-maker. It is designed for complex multi-level governance: strategic decisions that cross functions and where the confusion between giving an opinion, vetoing and deciding is the main source of paralysis.

The criterion for choosing between the two: DACI for the cross-cutting everyday flow; RAPID when there are legitimate vetoes to make explicit and several levels involved. And a criterion for using both: formalise recurrent or contentious decisions; do not bureaucratise the trivial ones. A framework applied to everything ends up applied to nothing.

8.3 Delegation poker: calibrating the level of delegation

The previous frameworks assign roles per decision; Management 3.0's delegation poker calibrates the degree of delegation between leader and team. Its seven levels run from "I decide and tell" (1) to "I delegate completely" (7), passing through selling the decision, consulting before deciding, agreeing jointly, advising without deciding and asking afterwards. The practice is a game: for each decision in the inventory, leader and team simultaneously reveal the card of the level each believes to be in force, and the discrepancies —which always appear— are discussed until the explicit level is agreed.

What the game reveals. The value of delegation poker lies not in the final number but in the discrepancies: when the leader believes they are at level 6 and the team perceives a 3, that distance is exactly the implicit architecture of chapter 1, measured. Repeating the calibration periodically turns the progressive transfer of authority of chapter 3 into something observable.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Assign decision rights with DACI or RAPID to a real case, choosing the appropriate framework and justifying the choice.
- Distinguish the parts of contributing, vetoing, deciding and being informed, and explain why confusing them paralyses cross-cutting decisions.
- Facilitate a delegation poker session on the team's decision inventory and turn the discrepancies into agreed levels.
- Apply the criterion of proportionality: which decisions deserve a formal framework and which do not.

Common mistakes and antipatterns

The matrix for everything. Requiring DACI or RAPID even to choose the meeting room. Bureaucratising the trivial discredits the framework exactly where it is needed.

The choral Approver. Spreading approval across five people "so that nobody is offended". A framework with several approvers reproduces the problem it came to solve: the decision has no owner again.

Roles assigned, behaviour unchanged. Publishing the matrix and carrying on deciding as always. The framework is only worth something if somebody invokes it when it is breached, starting with the Driver.

Confusing Input with Agree. Giving a de facto veto to somebody who was only meant to be consulted, in order to avoid conflict. It is the fast lane to the permanent consensus the next chapter analyses.

Further reading

· Atlassian — DACI

· Bain — RAPID

· Management 3.0 — Delegation Poker

BLOCK 4 · DECISION-RIGHTS FRAMEWORKS AND RECORD-KEEPING

Chapter 9. Consensus, consent and the advice process · CONSOLIDATING

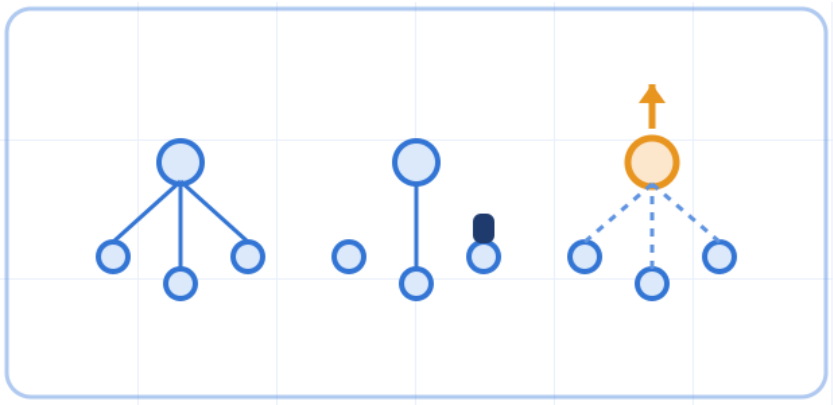
Each mechanism has a cost and a speed: the craft lies in matching mechanism to type of decision.

Introduction

When a team decentralises, it soon discovers that "deciding together" is not a mechanism but the absence of one. The inherited pairing —either the boss decides or the team votes— leaves out the more interesting options. This chapter presents the three alternatives that are entering the standard repertoire of agile teams, coming from sociocracy and from self-managed organisations, and the criterion for choosing between them.

9.1 Consensus as against consent

Consensus requires everybody to agree with the proposal: it is the most legitimising mechanism and, by far, the most expensive. Its failure mode is well known: the proposal is filed down until nobody hates it —and by then nobody wants it— or it is postponed indefinitely. Sociocratic consent reverses the question: not "do you all agree?" but "does anybody have a reasoned objection?". A reasoned objection is not a preference or a discomfort: it is an argument that the proposal will damage the team's capacity to fulfil its purpose.



Three mechanisms: consensus (everybody agrees), consent (absence of a reasoned objection) and the advice process (one person decides after consulting).

The difference looks subtle and is enormous: the space of "I have no objection" is much wider than that of "I agree", so consent allows far more to be decided, faster, while keeping the safeguard that matters: if something is going to do damage, whoever sees it is obliged to say so, and the objection is integrated into the proposal instead of blocking it.

9.2 The advice process

The advice process, coined by Dennis Bakke at AES and popularised by Frederic Laloux in Reinventing Organizations, goes a step further: whoever spots the problem decides —anybody, with no need for a role— with two prior obligations: to seek advice from those who will be affected and from those with relevant experience. The advice is genuinely listened to, but it is not binding: the decision and its responsibility stay with whoever makes it.

It is the fastest of the three mechanisms and the one that demands the most trust: it works where the two pillars of chapter 4 are in place and the culture tolerates somebody deciding "without permission" having consulted. In exchange, it produces exactly the dynamic block 2 is after: the decision is born where the information about the problem was born.

9.3 Matching mechanism and decision

No mechanism is superior in the abstract; each buys a different combination of speed, legitimacy and safety. The matching criterion uses the reversibility of chapter 6: for reversible decisions, the advice process or a direct decision by the owner according to the map —the cost of being wrong is low and speed is worth more than legitimacy. For decisions with appreciable collective impact, consent: fast but with a safety net of objections. Consensus is reserved for the few identity-defining and irreversible decisions where unanimous commitment is part of the result (team values, changes of purpose).

Rule of thumb. The more reversible the decision, the lighter the mechanism. If the team is voting on something that can be undone in a week, the mechanism has been badly chosen; if somebody decides alone on something irreversible that affects everybody, so has it.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the differences between consensus, consent and the advice process, including what a reasoned objection is.
- Choose a mechanism for a specific decision in a justified way, using reversibility and the scope of the impact as criteria.
- Facilitate a consent round: present the proposal, collect reasoned objections and integrate them.
- Apply the advice process to a decision of their own: identify those affected and the experts, collect their advice and decide taking on the responsibility.

Common mistakes and antipatterns

Consensus by default. Using the most expensive mechanism for everything, out of inertia or fear of conflict. The result is deliberative paralysis: teams that talk a great deal and decide little.

Objections that are preferences. Accepting any "I don't like it" as an objection. Consent only works if the team learns to tell argued damage from personal taste.

Advice process without advice. Keeping the comfortable half of the mechanism: deciding without genuinely consulting those affected and the experts. Without the obligation to seek advice, it is not an advice process: it is a unilateral decision with a brand name.

A single mechanism as identity. Turning one mechanism into a cultural badge ("here everything is decided by consent") and ignoring the matching. The mechanism serves the decision, not the other way round.

Further reading

· Sociocracy For All — consent decision-making

· Comparing decision-making methods (advice process)

BLOCK 4 · DECISION-RIGHTS FRAMEWORKS AND RECORD-KEEPING

Chapter 10. Decision records: ADRs and lightweight formats · ESTABLISHED

What was decided, what alternatives were ruled out and what was known at the time: the record that makes it possible, six months later, to judge fairly.

Introduction

In 2011, Michael Nygard published a brief note proposing that "architecturally significant" architecture decisions be documented in short text files alongside the code: Architecture Decision Records. The format —context, decision, status, consequences— was deliberately minimal, and precisely for that reason it prospered: today ADRs are widely recommended practice in software and their pattern has jumped to any field of decision.

10.1 The minimum viable format

Stripped of what is specific to architecture, a decision record useful for any field fits into three sections: what was decided (the decision in one sentence, with a date and an owner), what alternatives were ruled out (and why, one line each) and what was known at the time (the context, the constraints and the unknowns accepted). ADRs add the status (proposed, accepted, superseded) and the expected consequences, which make the record something that can be checked.

The hygiene rules of the original format are still the right ones: records numbered, immutable —a reversed decision is not deleted: it is marked as superseded by the new one— and kept where the work happens, not in a separate repository nobody visits. A decision record should cost ten minutes; if it costs more, the format has put on weight.

10.2 Why it is not bureaucracy

The reflex objection —"more paperwork"— confuses the record with the procedure. The record does not add steps to the decision: it photographs the one already made. And that photograph is the only thing that makes it possible, six months later, to tell a bad decision from a reasonable decision with bad luck: without knowing what was known then, every review is a verdict delivered with Monday's newspaper in hand, in which any decision that turned out badly looks obviously stupid and any that turned out well, obviously brilliant.

The six-month test. Before arguing about whether keeping records is bureaucracy, run the test: pick a technical or organisational decision from six months ago that generates complaints today, and ask which alternatives were weighed and what was known at the time. If nobody can answer with any confidence, the team is not in a position to learn from its decisions — only to blame itself for them.

There is a second, less cited dividend: writing the record improves the decision in progress. Listing the alternatives ruled out forces you to have considered them; making the accepted unknowns explicit turns them into signals to watch. The record is the written version of the "I intend to..." of chapter 3: complete thinking before acting. And, as chapter 6 anticipates, keeping a record is part of the obligatory treatment of every one-way door.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Draft a minimum decision record —decision, alternatives ruled out, known context— in under ten minutes.
- Apply the hygiene rules of the format: numbering, immutability and proximity to the work.
- Define the criterion for which of the team's decisions deserve a record, starting with the irreversible and the repeatedly disputed ones.
- Use existing records to answer, when a decision is questioned, what was known and what was ruled out at the time.

Common mistakes and antipatterns

The novel-length record. Three-page templates with fifteen sections. The cost kills the habit; the minimum format exists because only the minimum survives day-to-day work.

Record and bury. Keeping records in a space nobody consults. The record lives where the work lives: next to the code, on the team's page, linked from the decision map.

Rewriting history. Editing old records so that the decision looks better informed than it was. Immutability is not purism: it is what makes the record credible as memory.

Recording everything or recording nothing. Both extremes kill the practice. The proportionality criterion of chapter 8 applies here too: irreversible and contentious ones yes; trivial ones, no.

Further reading

· Nygard — Documenting Architecture Decisions

· adr.github.io — resources and templates

BLOCK 4 · DECISION-RIGHTS FRAMEWORKS AND RECORD-KEEPING

Chapter 11. Decision retrospectives · CONSOLIDATING

Evaluate the process, not the outcome: judgement also needs its inspect-and-adapt loop.

Introduction

Agile retrospectives regularly review what we did and how we coordinated. They almost never review how we decided. It is a curious omission in a culture built on inspection and adaptation: the decision system —the piece this whole topic is trying to design— is left outside the only mechanism that could deliberately improve it. This chapter presents the practice that closes that gap.

11.1 The practice: process, not outcome

The format is simple: take two or three decisions from the last quarter —ideally with their record from chapter 10 in front of you— and evaluate the process, not the outcome, with three questions: did we have the information it was reasonable to have at the time? Did the person who should have decided decide, according to the map? Did we take the appropriate amount of time for the type of door it was? The answers feed concrete adjustments: move a decision on the map, change the mechanism, make undoing cheaper.

The insistence on process has a basis: the outcome of a decision mixes the quality of the process with luck. An excellent decision can turn out badly and a reckless one can turn out well. Evaluating by outcomes teaches the team to be lucky, which cannot be trained; evaluating by process teaches it to decide better, which can.

11.2 The traps: hindsight bias and outcome bias

Two biases systematically sabotage this practice if they are not neutralised. Hindsight bias makes it seem, once the ending is known, that what was going to happen "was obvious": memory rewrites the uncertainty of the time. Outcome bias leads to judging the quality of the decision by its result: the one that turned out well was a "good decision", the one that turned out badly, "bad", regardless of what was known when it was made.

The main antidote to both is the record of chapter 10: the photograph of what was known and what was ruled out, taken before the ending was known, prevents memory from cheating. Hence the two practices go together: without records, the decision retrospective degenerates into the very verdict-with-Monday's-newspaper it was meant to avoid. An attentive facilitator adds the control question: "with what we knew then, what would we have said about this decision?".

The complete loop. Inspection and adaptation applied to the decision system itself: the map (chapter 5) makes the architecture visible, the frameworks and mechanisms (chapters 8 and 9) redesign it, the record (chapter 10) documents it and the decision retrospective improves it. With AI shifting the bottleneck towards judgement (block 6), deliberately training that judgement stops being optional.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Facilitate a decision retrospective on two or three real cases, evaluating available information, ownership and timing.
- Distinguish the quality of the process from the quality of the outcome, and explain why the former is what is evaluated.
- Recognise and neutralise hindsight bias and outcome bias using decision records as evidence.
- Turn the conclusions of the retrospective into concrete adjustments to the map, the mechanisms or reversibility.

Common mistakes and antipatterns

The outcome tribunal. Turning the session into a hunt for whoever is to blame for the decision that turned out badly. It guarantees that nobody will decide anything risky again, and that the next retrospectives will be exercises in self-defence.

Reviewing failures only. The decisions that turned out well with a bad process are the most dangerous: they teach recklessness. Reviewing a success with a dubious process vaccinates against outcome bias.

A retrospective without records. Evaluating from memory what was known a quarter ago. Memory, contaminated by the ending, will declare obvious what was uncertain.

Conclusions without adaptation. Diagnosing the same problems quarter after quarter without touching the map or the mechanisms. Inspection without adaptation is the useless half of the loop.

Further reading

· Fowler — Architecture Decision Record

· Kozyrkov — Introduction to Decision Intelligence

BLOCK 5 · DECIDING UNDER UNCERTAINTY

Chapter 12. The cost of waiting versus the cost of being wrong · ESTABLISHED

If you wait until you have all the information, you are not deciding: you are confirming.

Introduction

The previous blocks design who decides and with what safeguards. This block addresses the when and the how under conditions of uncertainty, which is the normal state of agile work. It starts by dismantling the most deeply rooted reflex: when in doubt, ask for more data. The reflex looks prudent and is often just a polite way of not deciding.

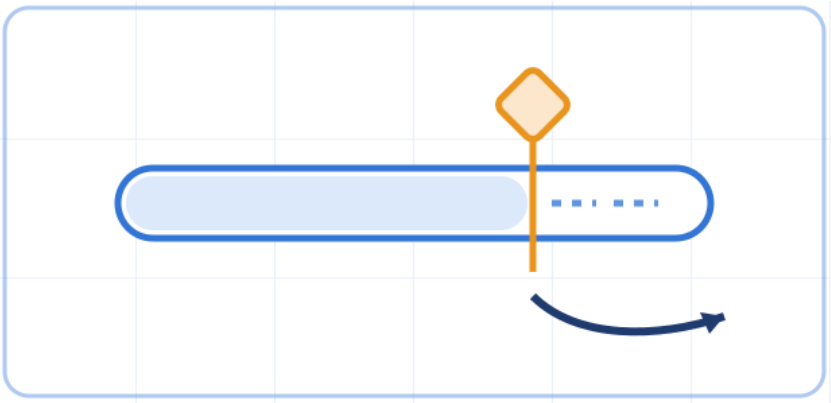
12.1 The right question

The usual question —"do we have enough data?"— has no answer: there is always room for one more analysis. The useful question is a different one: which costs more, waiting or being wrong? It is a comparison of two costs, both of them estimable. The cost of being wrong depends on reversibility (chapter 6): at a two-way door, being wrong costs the toll of going back. The cost of waiting is more insidious because it appears in no report: opportunities that expire, teams blocked downstream, competitors who have already moved, and the additional analysis itself, which also costs.

Put like that, Bezos's conclusion stops sounding reckless: if you are good at correcting course, being wrong may cost less than you think, whereas being slow is going to cost dearly for certain. Decision speed is not a character trait: it is the result of having made the comparison.

12.2 The 70% heuristic

The practical rule this reasoning distils: most decisions should be made with around 70% of the information you would like to have. If you wait for 90%, in most cases you are being slow. The remaining 30% does not come free with more analysis: it is bought more cheaply by acting, because action generates the information analysis can only estimate — it is the logic of the probe that chapter 14 formalises for complex domains.



With 70% of the information you would like, decide: the remaining 30% is bought more cheaply by acting than by analysing.

The heuristic comes with its condition of application, and it is worth making it explicit: it is conditional on reversibility. For reversible decisions —which are the majority— waiting almost always costs more, and 70% is even generous. For one-way doors, the calculation changes: the

additional deliberation is precisely the deliberate slowing down chapter 6 prescribes. The mistake lies not in being fast or slow, but in applying the speed of one type of door to the other.

Deciding is not confirming. When the information reaches the point where the choice is obvious, there is no longer a decision: there is confirmation. Deciding is, by definition, choosing with uncertainty remaining. Whoever is only comfortable deciding with certainty never decides: they wait for reality to decide and then sign the minutes.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Apply the waiting-versus-being-wrong question to a real case, estimating both costs explicitly.
- Use the 70% heuristic conditional on reversibility: aggressive at two-way doors, suspended at one-way doors.
- Identify the hidden costs of waiting in a specific flow: downstream blockages, opportunities that expire, the cost of the analysis itself.
- Recognise analysis paralysis and reframe it as an implicit decision not to decide, with its corresponding cost.

Common mistakes and antipatterns

The definitive piece of data. Postponing the decision "until we have the report that clears it up". The report clarifies the past; the relevant uncertainty is usually in the future, and only action reduces it.

70% as a universal excuse. Invoking the heuristic to rush one-way doors. The heuristic comes with its reversibility condition; without it, it is just haste with an authoritative quotation.

Deciding fast without correcting fast. The forgotten half of the reasoning: decision speed is only safe if the detection and correction of the error are fast too. Without a correction loop, deciding at 70% is simply betting.

Counting the visible cost only. Comparing the cost of being wrong (visible and attributable) with a cost of waiting that is assumed to be zero. Waiting always costs; it is just that its bill has nobody's name on it.

Further reading

· Bezos — 2016 letter to shareholders

· DHR Global — Talent Trends 2026

BLOCK 5 · DECIDING UNDER UNCERTAINTY

Chapter 13. The OODA loop and the primacy of Orient · ESTABLISHED

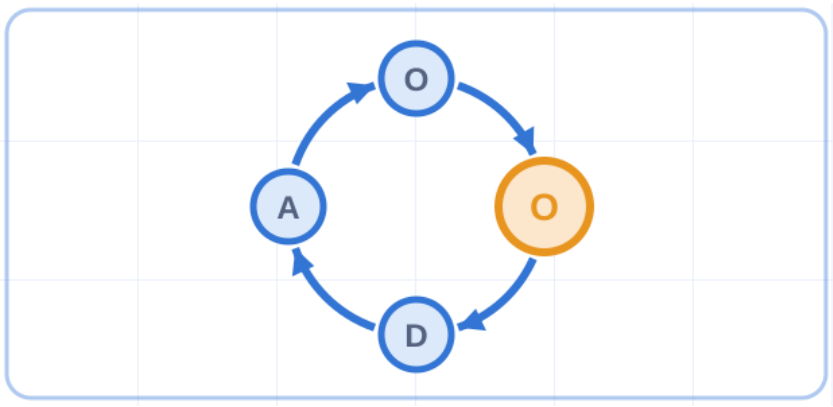
The winner is not whoever decides better in the abstract, but whoever iterates the complete loop faster — and the decisive phase is not Decide but Orient.

Introduction

John Boyd, fighter pilot and strategist of the United States Air Force, never published a book: his thinking is known through his briefings, his slides and the secondary sources that have distilled it. That circumstance explains why his most famous idea, the OODA loop —Observe, Orient, Decide, Act—, often circulates in simplified form: "whoever goes faster wins". The full version is considerably more interesting, and it describes agile work better than many texts written to describe it.

13.1 The loop as a description of agile work

The loop models how an agent —pilot, team, organisation— interacts with a changing reality: it observes what happens, orients what it has observed (interprets it with its experience, its context and its mental models), decides and acts; the action changes reality and the cycle starts again. Boyd's competitive thesis: the winner is not whoever decides better in the abstract, but whoever iterates the complete loop faster, because they operate on a fresher reading of reality while the adversary responds to a situation that no longer exists.



The OODA loop: the decisive phase is not Decide but Orient — the filter through which what is observed is interpreted.

The agile cadence is an institutionalised OODA loop: the sprint observes (feedback on what was delivered), orients (refinement, review), decides (planning) and acts (building), and starts again. Shortening the loop —more frequent deliveries, earlier feedback— is the operative translation of Boyd's advantage.

13.2 The nuance usually lost: Orient rules

A hasty reading puts the weight on Decide or on raw speed. Boyd insisted on the opposite: the decisive phase is Orient, the filter of experience, culture, training and mental models through which what is observed is interpreted. Orientation determines which data are considered signal and which noise, which options are even conceived and what is then decided almost automatically. Two teams look at the same dashboard and see different things: they do not observe differently; they orient differently.

From this follows an uncomfortable warning for the cult of speed: going very fast with the wrong orientation is not agility — it is being wrong more often per unit of time. Orientation can be worked on: teams with real diversity of mental models, direct exposure to reality (talking to users, seeing production) and explicit review of their own assumptions —the decision retrospective of chapter 11 is, among other things, orientation maintenance.

A preview of block 6. It is worth holding on to this nuance because AI is going to stretch it: generative AI spectacularly accelerates Observe and Act, and does not solve Orient. The loop will turn faster than ever around its most human phase. That asymmetry is the starting point of chapter 15.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Describe the four phases of the OODA loop and map them onto the cadence and events of agile work.
- Explain the primacy of Orient and diagnose a real bad decision by identifying the orientation phase as its origin.
- Propose practices that improve a team's orientation: diversity of mental models, direct contact with reality, review of assumptions.
- Argue why loop speed without quality of orientation does not constitute agility.

Common mistakes and antipatterns

OODA as a speed slogan. Reducing the loop to "decide faster than the other side". Without the primacy of Orient, the recipe accelerates successes and mistakes alike.

Inherited, invisible orientation. Operating for years with the same mental models without reviewing them. Orientation expires silently: what was signal becomes noise and nobody updates the filter.

Orientation monoculture. Homogeneous teams that interpret everything the same way. Convergence feels like harmony and is fragility: a single blind spot shared by everybody. Chapter 22 shows how AI can aggravate it.

Observing as a substitute for orienting. Accumulating dashboards and metrics in the hope that the data will interpret themselves. More observation with the same orientation produces the same reading, with more decimal places.

Further reading

· OODA loop — overview

· The Tao of Boyd — the complete version of the loop

BLOCK 5 · DECIDING UNDER UNCERTAINTY

Chapter 14. Cynefin as a decision contextualiser · ESTABLISHED

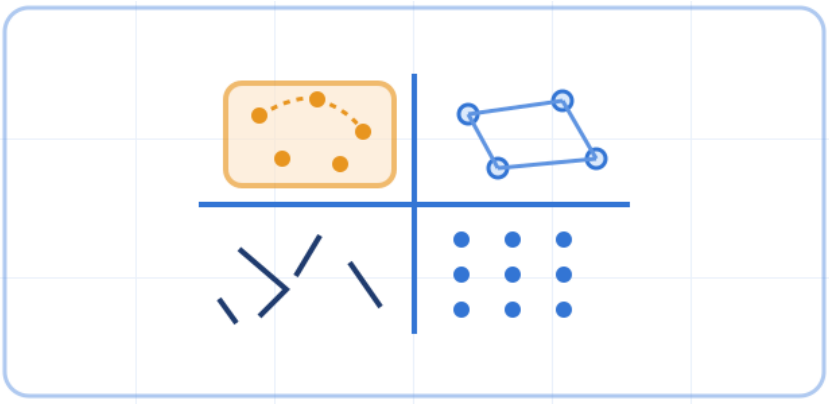
Match the type of context —clear, complicated, complex, chaotic— with the mode of decision that corresponds to it.

Introduction

Before deciding how to decide, there is a prior question: what type of context am I in? The Cynefin framework, created by Dave Snowden and presented to a wide audience in the article "A Leader's Framework for Decision Making" (Harvard Business Review, 2007, with Mary Boone), exists to answer it. Its premise: management approaches fail not because of poor execution but because they are applied in the wrong domain, since they assume a degree of order and predictability the context does not have.

14.1 The four domains and their modes of decision

Cynefin distinguishes four operational domains. In the clear domain, the cause-effect relationship is obvious to anybody: you sense, categorise and respond with established good practice. In the complicated one, the cause-effect relationship exists but requires analysis or expertise: you sense, analyse and respond; it is the realm of the expert and of the several possible correct answers. In the complex one, the cause-effect relationship is only recognised in retrospect: there is no analysing the correct response because the system changes when you touch it; the mode is probe-sense-respond: safe-to-fail experiments from which patterns emerge. In the chaotic one there is no perceptible cause-effect relationship: you act first to stabilise, then sense and respond.



Four domains, four ways of deciding; in the complex one —patterns that only emerge on probing— is where the sprint lives.

14.2 The characteristic mistake and the practical use

The most frequent and most expensive domain mistake in knowledge work is treating the complex as complicated: convening more analysis, more experts and more planning for a problem whose behaviour is only revealed by experimenting. The symptom is recognisable: ever more sophisticated analyses that age before they are finished, and recurrent surprise when the system "does not respond the way the plan said". The right answer was not a better plan: it was a cheaper and faster probe.

Hence the practical use for an agile team: the sprint is a probe in the complex domain. Delivering a real increment and observing the system's response —users, market, production— is exactly the probe-sense-respond the domain demands. This also sets a boundary: for genuinely complicated problems (sizing a database, complying with a regulation), expert analysis remains the correct mode, and dressing it up as an experiment is waste. Cynefin does not say everything is complex; it says you have to look before choosing the mode.

Connection with the map. Cynefin adds a useful column to the decision map of chapter 5: the domain. Clear decisions are delegated with good practice; complicated ones, to whoever has the expertise; complex ones, to the team with the capacity to probe; chaotic ones, to whoever can act now. The domain also informs the conversation of chapter 12: in the complex, the "remaining 30%" of information literally does not exist yet — only the probe generates it.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Place a real problem in its Cynefin domain and derive the corresponding mode of decision.
- Detect the mistake of applying complicated-domain analysis to complex-domain problems, and redirect it towards safe-to-fail probes.
- Explain the sprint as a probe in the complex domain and design increments that work as experiments.
- Use the domain as an additional criterion of the decision map when assigning owners and mechanisms.

Common mistakes and antipatterns

Everything is complex. Using Cynefin to justify universal improvisation. The clear and complicated domains exist, and in them good practice and expert analysis beat the experiment.

Analysis where a probe was called for. The characteristic mistake: months of study for a problem a two-week test would have illuminated better and more cheaply.

Probes that are not safe-to-fail. Calling an irreversible bet an experiment. The probe of the complex domain is, by definition, a two-way door (chapter 6); if its failure does serious damage, it was not a probe.

Classifying once and for all. Problems migrate between domains: the complex, once understood, becomes complicated and then clear. Reviewing the classification is part of orientation maintenance (chapter 13).

Further reading

· Snowden and Boone — A Leader's Framework for Decision Making

BLOCK 6 · AI COMPRESSES THE LOOP

Chapter 15. Asymmetric compression of the OODA loop · CONSOLIDATING

AI makes Observe and Act cheaper; the bottleneck shifts to exactly the two middle phases: to judgement.

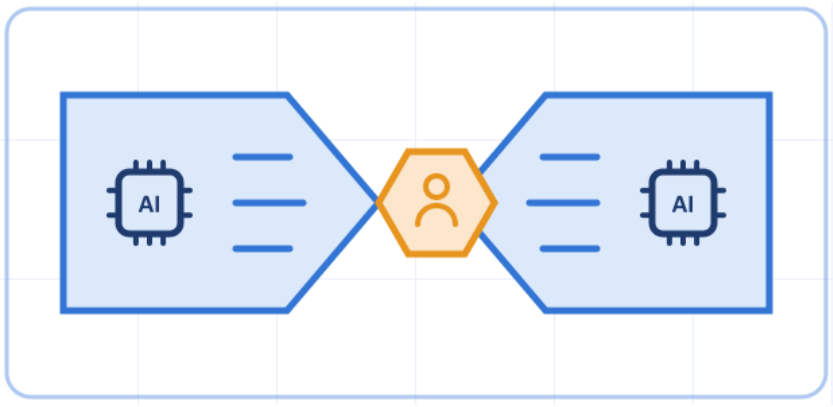
Introduction

Chapter 13 left a warning on the table: AI was going to stretch the OODA loop at its most human phase. This chapter develops how. Generative AI does not accelerate the loop uniformly: it compresses it asymmetrically, and understanding that asymmetry is the key to the whole AI layer of this topic. Without it, AI adoption is managed as a productivity problem; with it, it is managed as what it is: a shift of the bottleneck towards judgement.

15.1 What AI makes cheaper: Observe and Act

On the Observe side, AI summarises documents, analyses volumes of data previously out of reach, spots patterns, monitors signals and prepares syntheses in minutes. What used to cost weeks of collecting and digesting costs a conversation. On the Act side, it generates code, documents, prototypes, campaigns and configurations at a rate no human team matches: the distance between deciding something and having a first materialisation of it has shrunk by orders of magnitude.

The net result is that the loop turns faster than ever. But the two middle phases —Orient (interpreting what has been observed) and Decide (choosing with judgement)— have not become cheaper at the same rate, because they depend on the context, the values, the responsibility and the mental models of whoever decides. The mechanical consequence: the system's bottleneck shifts exactly there.



Asymmetric compression: observing and acting widen with AI; orienting and deciding become the bottleneck.

15.2 The new constraint: validating faster than things are generated

In the theory of constraints, optimising away from the bottleneck does not improve the system: it accumulates inventory in front of the constraint. Applied here: every additional improvement in generation (more drafts, more proposals, more code) that does not come with judgement capacity simply piles up material waiting to be validated. The new constraint of the knowledge value chain is validating faster than things are generated — and validating is an activity of Orient and Decide, not of Act.

The adoption data confirm that this is no longer a projection: according to Deloitte's 2026 human capital trends study, 60% of executives regularly use AI to support their decisions, and Gartner projects that by 2027 half of business decisions will be augmented or automated by agents. The operative question for a team is not whether AI will take part in its decisions, but whether its judgement capacity is growing at the pace of its generation capacity.

The asymmetry in one sentence. AI has made producing answers cheaper; it has not made knowing which ones to accept cheaper. All the organisational design of blocks 7 and 8 —real oversight, deliberate friction, hybrid map, trained judgement— is the answer to that sentence.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the asymmetric compression of the OODA loop: which phases AI makes cheaper and why Orient and Decide do not become cheaper at the same rate.
- Identify in a real workflow where generated material is accumulating awaiting validation.
- Apply the logic of the constraint: direct improvements towards judgement capacity when this is the bottleneck, instead of continuing to optimise generation.
- Use current adoption data to argue the urgency of designing the judgement layer, not only the generation layer.

Common mistakes and antipatterns

Measuring AI adoption by volume generated. Celebrating how many drafts, lines of code or proposals the AI produces. Without equivalent validation capacity, that volume is inventory in front of the bottleneck, not value.

Accelerating generation to compensate for slow decisions. The intuitive reflex and exactly the wrong one: more generation with the same judgement enlarges the queue and the pressure to approve without reviewing.

Treating validation as a formality. Assigning the review of what has been generated to whoever has a gap in their diary, instead of to whoever has the judgement. Validation is the scarce activity; it deserves the people with the most judgement, not the most available.

Assuming AI will orient too. Expecting the same system that generates to say what to accept. It can help (chapter 17), but outsourcing orientation entirely is the direct route to the automation bias of chapter 18.

Further reading

· Deloitte — AI and the future of human decision-making

BLOCK 6 · AI COMPRESSES THE LOOP

Chapter 16. The new economics of leadership · CONSOLIDATING

When producing is cheap, value lies in deciding well what to produce: whoever decides becomes the limiting factor of the value chain.

Introduction

The compression of the previous chapter has an economic reading that changes where the value of leadership lives. For decades, producing was expensive: value lay in executing well what had been decided, and the entire organisation was designed around the efficiency of execution. When producing becomes cheap —and AI is making it cheaper at great speed— value migrates upstream: to deciding well what to produce, what to validate and what to discard. This chapter runs through the three consequences of that migration.

16.1 Whoever decides is the limiting factor

If generation is abundant and judgement scarce, elementary economics says the price of judgement goes up. In teams working with AI this can already be seen in the evolution of roles: the Product Owner evolves towards the AI Product Manager profile —whoever frames problems, specifies intent and validates results for an almost unlimited production capacity— and becomes the new strategic bottleneck. The guide Scrum in teams with AI develops this evolution in detail; what matters here is its design reading: an organisation that does not identify and reinforce its points of judgement will see its whole generation capacity waiting in a queue in front of them.

The same logic reorders the priorities of people development: the competences of framing, specifying intent (chapter 4) and validating go from desirable to critical, while some of the competences of pure production lose relative weight. DDI's trends report for 2026 names it "AI fluency": the ability to question AI outputs, identify their biases and combine machine efficiency with human judgement, empathy and context. "Human + AI" leadership is, according to that report, the central trend of the year.

16.2 Decision saturation

The second consequence is less intuitive: abundance not only displaces value, it also saturates. When generating options is cheap, the problem stops being a lack of information and becomes the volume of competing decisions: every generated proposal is a pending micro-decision (accept it? review it? discard it?), and teams begin to experience decision saturation: too many open choices at once, with the cognitive cost and the procrastination that produces.

The answer is not heroic but structural: filtering and assigning ownership become a function of leadership. Filtering: deciding which decisions are not even opened (framings, prior criteria, relevance thresholds). Assigning ownership: making sure every open decision has an owner and a mechanism according to the map of chapter 5, so that the queue is managed by design and not by anxiety. Saturation is the definitive argument in favour of explicit architecture: with dozens of daily micro-decisions, the implicit architecture simply collapses.

Leadership as the management of scarcity. In the previous economy, the leader managed scarce production capacity. In the new one, they manage scarce judgement: where it is applied, how it is protected from saturation and how it is trained (chapter 23). Same craft, different resource.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the migration of value from execution to decision and its consequences for the design of roles and teams.
- Identify the limiting points of judgement in a specific value chain and propose how to reinforce them.
- Recognise the symptoms of decision saturation in a team and apply the two structural responses: filtering and assignment of ownership.
- Define what "AI fluency" means in their context and translate it into criteria for the team's professional development.

Common mistakes and antipatterns

Reinforcing execution in the middle of the value migration. Continuing to invest only in production capacity when the bottleneck is already judgement. It is optimising the abundant part of the system.

The universal validator. Concentrating all validation in one person (often the PO) without redesigning the flow. The new strategic bottleneck needs the treatment of chapters 5 and 8: map, rights and delegation, not overtime.

Confusing saturation with lack of commitment. Interpreting procrastination in the face of dozens of open decisions as a problem of attitude. It is a problem of volume without filter or ownership: it is solved by design, not by pressure.

AI fluency as a tools course. Reducing the competence to knowing how to use AI applications. The fluency that matters is fluency of judgement: questioning outputs, detecting biases, deciding when not to follow the recommendation.

Further reading

· DDI — Leadership Trends 2026

· Scrum in teams with AI — Skill Arena

BLOCK 6 · AI COMPRESSES THE LOOP

Chapter 17. AI to open options, not to close them · CONSOLIDATING

AI widens the menu; the team chooses the dish.

Introduction

There are two ways of bringing AI into a decision, and although they use the same tool they produce opposite effects on judgement. The first treats it as an oracle: you put the problem to it and adopt its answer. The second treats it as a generator of alternatives and a devil's advocate: you ask it to widen what the team is considering and to attack what the team already believes. This chapter develops the second, which is the one that improves judgement instead of replacing it.

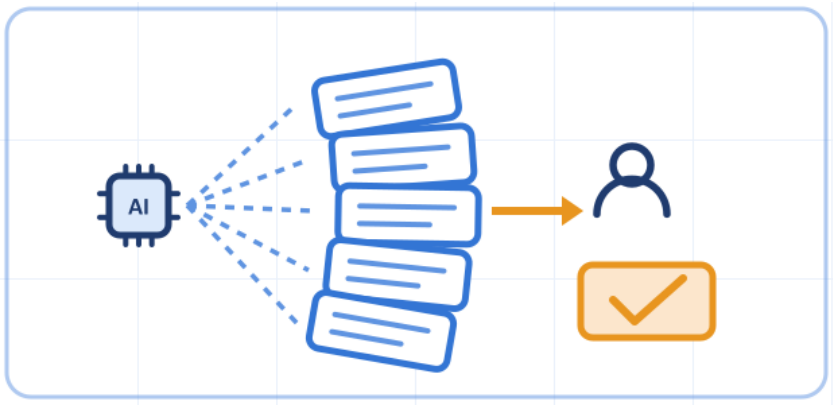
17.1 The oracle: a machine for manufacturing agreement

Used as an oracle, AI delivers a single answer, well written, self-assured. That fluency has a documented psychological effect: it is hard to disagree with an impeccable text, especially under time pressure. The team does not deliberate: it agrees. The decision looks as if it was made with AI and has in fact been made by the AI, with the team in the role of notary — the exact prelude to the automation bias that opens block 7.

The problem is not that the oracle's answer is bad; it is often reasonable. The problem is what it does not show: the alternatives it did not generate, the assumptions it did not question, the uncertainty its tone does not convey. A decision is only as good as the space of options it was chosen from, and the oracle shrinks that space to one.

17.2 The option generator and the devil's advocate

The inverse use asks the AI for precisely what the oracle hides. As a generator of alternatives: "give me five different approaches to this problem, including at least one conservative, one aggressive and one we would not have thought of". As a devil's advocate: "what are we assuming without realising it?", "how could this option go wrong?", "argue against what we have decided". AI is exceptionally good in both roles, because generating variations and objections is exactly the type of task that generation makes cheaper (chapter 15).



AI widens the menu of options and subjects them to criticism; the team chooses the dish.

Within this repertoire, the star practice is the premortem: before executing the decision, ask the AI to place itself in the future and narrate why it failed ("we are a year on and this decision turned out badly: tell us what happened"). The premortem turns the social discomfort of objecting —nobody wants to be the doom-monger of the meeting— into an external exercise with no personal cost, and AI makes it systematic and cheap. The skill AI applied to agile work introduces this competence; here it is integrated into the decision system: a systematic premortem for every one-way door (chapter 6) before crossing it.

Practical rule. AI widens the menu; the team chooses the dish. If, in an AI-assisted decision, you cannot point to at least two real alternatives that were considered and one serious objection that was discussed, you have not used AI to decide: you have used the decision to ratify the AI.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish oracle use from option-generator use, and identify which is happening in a specific decision.
- Formulate requests to AI that widen the space of options: diverse alternatives, hidden assumptions, arguments against.
- Facilitate an AI-assisted premortem before an irreversible decision and turn its findings into concrete safeguards.
- Apply the menu rule: verify that every AI-assisted decision leaves a trace of the alternatives considered and the objections discussed.

Common mistakes and antipatterns

The oracle with intermediate steps. Asking for five options and always choosing the one the AI presents first or with the best prose. The menu was decorative; the agreement, the same.

A pro forma devil's advocate. Generating the objections and discussing none. A premortem that changes nothing in the decision —not a safeguard, not a threshold, not a plan B— is due-diligence theatre.

Opening options for trivial decisions. Deploying the full repertoire to choose the colour of a button. The cost of the method should be proportional to the decision (chapter 8): the premortem is for one-way doors, not for all of them.

Forgetting that AI also converges. Assuming that asking a single model for options guarantees real diversity. The options of one model share its orientation; for major decisions, vary sources and models (chapter 22).

Further reading

· Deloitte — AI and the future of human decision-making

BLOCK 7 · HUMAN JUDGEMENT AND OVERSIGHT

Chapter 18. Automation bias: agreement is not judgement · CONSOLIDATING

Approving the AI's proposal without having been able to evaluate it feels exactly the same as deciding. But it is not deciding: it is signing off.

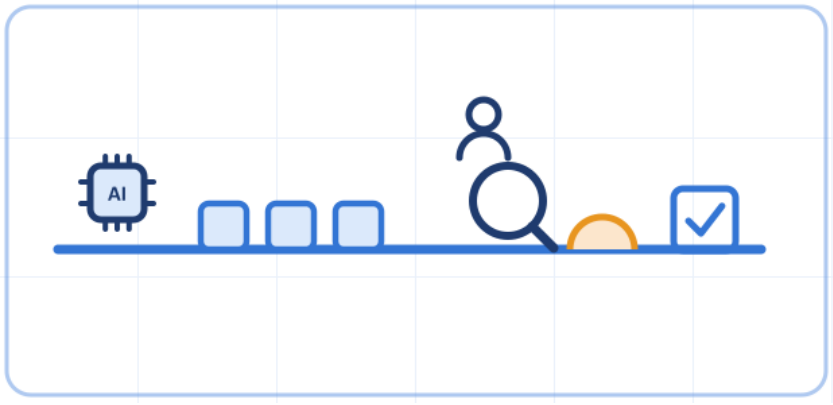
Introduction

Block 6 establishes that judgement is the scarce resource; this block deals with its main threat: that it degrades without anybody noticing. The mechanism is old —it was documented in aircraft cockpits and control rooms long before generative AI— but its scale is new: when every workflow incorporates a system that proposes, automation bias stops being a niche risk and becomes the first-line risk of teams that decide with AI.

18.1 The mechanics of the bias

Automation bias is the tendency to accept the proposals of an automated system without the evaluation we would apply to a human proposal. Its mechanics are statistically reasonable, and that is why it is so effective: when a system is right 90% of the time, reviewing every proposal looks like waste; the natural thing is to stop reviewing. And then the remaining 10% goes through without anyone looking at it — precisely the cases where the review contributed all of its value.

The subtle trap lies in the phenomenology: agreement looks very much like judgement. Approving the AI's proposal without having been able to evaluate it feels exactly the same as deciding —you read, you nod, you click approve— so that the degradation of judgement is invisible to whoever suffers it. It is not deciding: it is signing off. And a team can spend months signing off, convinced that it is deciding.



Real oversight: time and competence to review, and deliberate friction before irreversible decisions.

18.2 The three conditions of real oversight

Whether human oversight is real and not ritual depends on three conditions, all of them a matter of design. Time: if the volume of proposals exceeds the capacity for review, oversight is arithmetically impossible however much goodwill there is — it is the saturation of chapter 16 applied to review. Competence: only somebody who could have formulated the proposal, or who knows how to recognise its failure modes, can evaluate it; reviewing above your own competence is agreeing with extra steps. And the ability to dissent at no cost: if rejecting the AI's proposal requires justification

while accepting it does not, the system of incentives has already decided for everybody. If any of the three is missing, the review is theatre.

Design, not virtue. The three conditions are organisational, not moral: they are not solved by asking people to "review more carefully", but by sizing volumes, assigning competent reviewers and removing the asymmetry of cost between dissenting and agreeing. Automation bias is fought with decision architecture, which is precisely the craft of this guide.

18.3 The connection with the harness

This chapter has its technical mirror in Harness Engineering: the antipattern of exclusive faith in the human reviewer as the only safeguard on the work of agents. The harness responds with guides (feedforward) and sensors (feedback) that relieve human review of what can be verified mechanically, reserving judgement for what only judgement can evaluate. It is the same economics of scarce judgement as this block, implemented in infrastructure: the better the harness, the more real the oversight that remains can be.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the mechanics of automation bias and why agreement is subjectively indistinguishable from judgement.
- Audit an AI-assisted decision flow against the three conditions of real oversight: time, competence and dissent at no cost.
- Redesign a review point that has degenerated into ritual: size the volume, reassign competence, balance the cost of dissenting.
- Connect human oversight with the infrastructure of the harness: what to offload to guides and sensors and what to reserve for judgement.

Common mistakes and antipatterns

The decorative review. Keeping a human approval step that approves one hundred per cent of what it receives. A review that never rejects is not a safeguard: it is latency with a signature.

Blaming the person for the bias. Treating agreement as a lack of individual diligence. With forty proposals a day, the most diligent person in the world agrees; the problem is the design of the flow.

Reviewing above your competence. Assigning approval to whoever has the hierarchical authority but not the technical competence. It satisfies the organisation chart and empties the oversight.

Punishing dissent. Requiring detailed justification to reject what the AI proposes and none to accept it. The asymmetry of cost manufactures agreement on an industrial scale.

Further reading

· Harness Engineering — Skill Arena

· DDI — Hot Leadership Topics 2026

BLOCK 7 · HUMAN JUDGEMENT AND OVERSIGHT

Chapter 19. Accountability in AI-assisted decisions · EMERGING

Responsibility without a real capacity to review is not responsibility: it is a lightning rod.

Introduction

Who is accountable for a decision suggested by an AI? The easy answer —"the human who approved it"— settles the paperwork and dodges the problem. If that human approves forty suggestions a day because the flow leaves no room for more, their responsibility is a legal fiction: they are there to absorb the discharge when something fails, not because they could have prevented it. This chapter turns that discomfort into design requirements.

19.1 Real responsibility as against a lightning rod

The operative distinction: there is real responsibility when whoever answers for the decision had an effective capacity to evaluate it and to decide differently — the three conditions of chapter 18, met. There is a lightning rod when the structure assigns blame to a human point whose practical function is legal-decorative. The lightning rod is comfortable for everybody until the lightning strikes: then it turns out that the organisation did not have oversight, it had a fuse with a name attached.

The design consequence is twofold. First: size the volume of decisions to the real capacity for review — if capacity cannot grow, the volume that demands judgement must be filtered (chapter 16) or explicitly delegated with its safeguards (chapter 21). Second: make explicit which decisions require substantive human evaluation, distinguishing them from those approved by rule; mixing the two in the same queue is the recipe for a lightning rod.

19.2 The framework on its way: meaningful oversight and auditable authorisation

AI regulation is converging on the principle of meaningful human oversight: a human in the loop is not enough; they must be able to understand, question and reverse. What was an ethical debate is becoming a formal requirement, and the 2026 agent governance frameworks are giving it instrumentation: the World Economic Forum's playbook introduces the Agent Capability and Authorization Profile (ACAP), an authorisation profile per deployment that makes delegated decisions and actions auditable, enforceable and attributable throughout their life cycle — who authorised what, within which limits and under what oversight.

The honest design question. For each point of human approval in your flow: if this decision turns out badly, could this person reasonably have prevented it? If the answer is no, there are two honest options: give them the capacity (time, competence, volume) or take away the responsibility and redesign the point. Keeping the signature without the capacity is choosing the lightning rod knowingly.

19.3 What a team can do without waiting for regulation

Without waiting for the complete regulatory framework, a team can apply ACAP logic today on a small scale to its decision map: for each assisted or delegated decision, make explicit who authorises it, within which limits, who oversees it and how it is audited — which is the record of chapter 10

extended to the AI layer. The dividend is twofold: defensible accountability today and cheap adaptation when the regulation arrives.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Distinguish real responsibility from a lightning rod in a specific flow, applying the criterion of effective capacity to evaluate and to decide differently.
- Size the volume of decisions that demand judgement to the real capacity for review, and make explicit which require substantive human evaluation.
- Explain the principle of meaningful human oversight and the function of auditable authorisation profiles (ACAP).
- Document, for each assisted or delegated decision, who authorises it, within which limits and how it is audited, extending decision records to the AI layer.

Common mistakes and antipatterns

The universal signatory. Concentrating every approval in the AI flow in one figure in order to "have somebody responsible". It is the deliberate manufacture of a lightning rod, with paralysis or agreement as the only possible modes of operation.

Human in the loop as a checkbox. Inserting a human into the circuit for compliance, without giving them the capacity to understand or to reverse. It satisfies the letter of the requirement and betrays its purpose.

Responsibility without authority. Making somebody answer for decisions whose limits and configuration they do not control. Accountability demands symmetry: you answer for what you were able to govern.

Audit trails reconstructed after the fact. Documenting who authorised what only once something has failed. Traceability that did not exist before the incident is, at best, a novel.

Further reading

· WEF — AI Agents in Action (ACAP)

BLOCK 7 · HUMAN JUDGEMENT AND OVERSIGHT

Chapter 20. Deliberate friction as a safety mechanism · EMERGING

A certain amount of friction in irreversible decisions is a safety mechanism: if AI removes it, it has to be reintroduced by design. Not slowness: composure.

Introduction

Removing friction is the favourite sport of process improvement, and generally with good reason: friction is usually waste. But not all of it. A certain amount of friction in making irreversible decisions —the formality that forces you to wait, the second signature, the prior report— worked, whether we knew it or not, as a safety mechanism: time to think disguised as bureaucracy. AI, that tireless optimiser of flows, removes that friction along with the bad kind. This chapter is about how to reintroduce it on purpose, only where it protects.

20.1 Speed as a parameter, not as an achievement

The compression of the loop (chapter 15) also drags in decisions that should not be accelerated: when the whole flow runs at machine speed, the one-way door arrives at the same pace as the two-way ones, and is crossed with the same gesture. The classic rule of chapter 6 —irreversible decisions are deliberately slowed down— already allowed for chosen slowness; Bezos went so far as to jokingly call himself "chief slowdown officer" for one-way decisions. What is new is that the default speed is now set by AI, so the slowing down no longer happens on its own: it has to be designed. Speed stops being an achievement and becomes a design parameter, configured by type of decision.

20.2 Patterns of deliberate friction

The practical repertoire is short and effective. Mandatory pauses: an irreversible decision cannot be approved in the same session in which it is proposed; a night in between is the cheapest safety mechanism there is. A second human reviewer: for one-way doors, a second signature with real competence —not hierarchical— breaks the inertia of individual agreement. Proportional reflection periods: the greater the irreversibility, the longer the minimum interval between proposal and execution. And rate limiters in automated flows: thresholds that, when exceeded (amount, scope, irreversibility), take the decision out of the fast lane and return it to the circuit with judgement — the decision-making equivalent of circuit breakers in the financial markets.

Composure, not slowness. Deliberate friction does not defend slowness: it defends selective composure. A well-designed system is bimodal: extremely fast at two-way doors and deliberately calm at one-way doors. Whoever accuses it of being slow is looking only at the second mode; whoever wants everything fast is asking the system to cross its irreversible doors at a trot.

20.3 The cadence of judgement as a leadership craft

Managing the cadence of judgement —which decisions run, which wait and for how long— is a new part of the leadership craft, sibling to the one chapter 16 describes for filtering. Its tool is the decision map of chapter 5 with one more column: assigned speed. And its discipline is resisting the

cultural pressure that every wait is a failure: in decision architecture, some waits are the system working exactly as designed.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Identify, in a flow accelerated by AI, the irreversible decisions that have lost their protective friction.
- Apply the patterns of deliberate friction —mandatory pauses, second reviewer, proportional intervals, rate limiters— to the type of decision that warrants them.
- Configure speed as a parameter by type of decision in the team's map, justifying each assignment by reversibility.
- Defend selective composure to the organisation, distinguishing it from slowness and explaining its safety function.

Common mistakes and antipatterns

Uniform friction. Reintroducing formalities for everything, restoring the bureaucracy agility dismantled. Deliberate friction is surgical: only at one-way doors.

Pauses that get skipped. Establishing the reflection period and granting yourself an exception every time there is a rush, which is always. Optional friction is not a safety mechanism: it is a suggestion.

A second signature without a second look. Adding an approver who signs in fifteen seconds. It duplicates the ritual of chapter 18 instead of breaking it; the second signature is worth whatever its competence and its time are worth.

Confusing the speed bump with the wall. Designing frictions that in practice block the decision indefinitely. The aim is to decide calmly, not to avoid deciding: every friction comes with its deadline and its way out.

Further reading

· Bezos — irreversible decisions and deliberate slowness

BLOCK 8 · DESIGNING THE HYBRID HUMAN–AI ARCHITECTURE

Chapter 21. Delegating decisions to agents · EMERGING

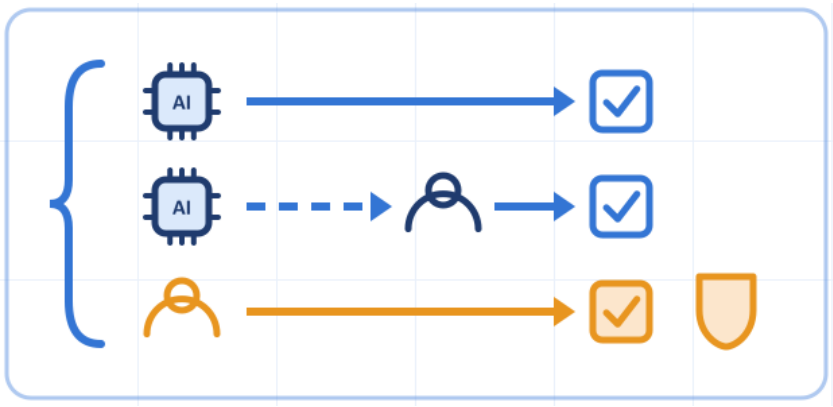
What the agent decides, what it proposes for approval and what is exclusively human: the team's decision map, extended.

Introduction

The last block of the guide introduces no new criteria: it extends those already built to a new type of member of the system. When an AI agent can make decisions —close an incident, adjust a parameter, merge a change—, the team's question is the same as in chapter 5, with one more lane: which decisions can the agent make, which work in propose-and-approve mode and which are exclusively human?

21.1 The three lanes of the hybrid map

The extended decision map organises each decision into one of three lanes. Delegation lane: the agent decides and executes; the human audits after the fact by sampling or through sensors. Proposal lane: the agent prepares the complete decision —analysis, alternatives, recommendation— and a human with the three conditions of chapter 18 approves or rejects. Human lane: decisions that are neither delegated nor prepared automatically; chapter 23 explains why some are protected even when they could be delegated.



Three lanes of the hybrid map: the agent decides, propose-and-approve, and protected decisions that are exclusively human.

What is notable is that the criteria for assignment are all already in this guide and apply unchanged. Reversibility (block 3): only two-way doors with a known cost of undoing enter the delegation lane; one-way doors live in the human lane with their deliberate friction. The two pillars (chapter 4): the agent receives authority over what it has demonstrated competence in —with a verifiable track record, not with faith— and always under explicit intent: the spec is its commander's intent. Explicit rights (chapter 8): the agent's authority is written down, with limits, just like that of any other member; an agent with implicit authority is the implicit architecture of chapter 1 running at machine speed.

21.2 The 2026 frameworks: delegation with authority, responsibility and trust

Delegation to agents moved in 2026 from improvisation to a discipline with formal frameworks. Google DeepMind's proposal on intelligent delegation formalises what organisational sense already intuited: delegating to an agent is not handing out tasks, but transferring authority and responsibility with a clear specification of roles and limits, clarity of intent and trust mechanisms —capability assessment, reputation, monitoring— between the parties, whether human or agent. In parallel, the World Economic Forum's playbook contributes the governance instrument: authorisation profiles (ACAP) that make each delegation auditable and enforceable (chapter 19). The convergence is remarkable: the leading frameworks for delegating to agents reproduce, formalised, the criteria of blocks 2 to 4 of this guide.

The technical implementation already exists. The guides (feedforward) and sensors (feedback) of the Harness Engineering harness are the technical implementation of an agent's decision rights: the guides delimit what it can decide before acting, the sensors verify what was decided afterwards. Whoever has a harness already has the infrastructure of the delegation lane; all they lack is the map that says what travels along it.

21.3 How to start: verifiable gradualism

The transfer of authority to agents follows the same progressive logic as chapter 3, accelerated by the advantage that everything is recorded. The prudent sequence: start in the proposal lane, where the agent demonstrates competence without risk; measure its track record with the sensors; promote to the delegation lane the reversible decisions where the track record supports it, with explicit limits and audit by sampling; and review the hybrid map in the decision retrospective (chapter 11), promoting and demoting lanes according to evidence. Delegating to agents is not an initial act of faith: it is a trust race with data.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Extend the team's decision map with the three hybrid lanes, assigning each decision with explicit criteria.
- Apply reversibility, demonstrated competence and explicit intent as conditions of entry to the delegation lane.
- Explain what the 2026 delegation frameworks add (transfer of authority and responsibility, trust mechanisms, authorisation profiles) over the handing out of tasks.
- Design an agent's progression from the proposal lane to the delegation lane with a verifiable track record, written limits and auditing.

Common mistakes and antipatterns

Delegating by capability, not by reversibility. Giving the agent everything it knows how to do. The criterion for entry is not whether it can, but whether the decision is reversible and its cost of error known; capability without reversibility is automated risk.

The agent with implicit authority. Letting the agent accumulate de facto decisions without anybody having written down its limits. It is the "it depends" of chapter 5 executing a thousand times an hour.

Skipping the proposal lane. Going from zero to full delegation with no track record. Trust in agents is built the same way as in people: with progressive and verifiable transfer, only here the data are free.

Delegation without sensors. Opening the delegation lane without the instrumentation that audits it. Delegating to an agent without sensors is not trust: it is blindness with a good press.

Further reading

· DeepMind — Intelligent AI Delegation

· WEF — AI Agents in Action (ACAP)

· Harness Engineering — Skill Arena

BLOCK 8 · DESIGNING THE HYBRID HUMAN–AI ARCHITECTURE

Chapter 22. The risk of cognitive re-centralisation · EMERGING

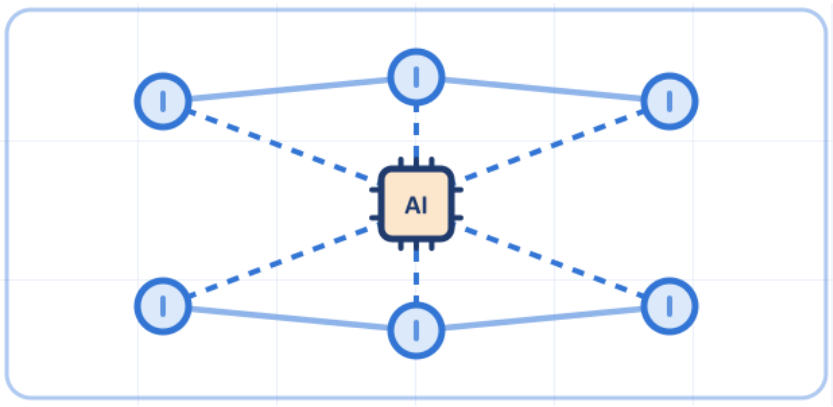
Decisions formally distributed, cognitively homogeneous: diversity of judgement was a property of the system that is not being protected.

Introduction

There is a paradox brewing in the organisations that have done the homework of this guide best: they decentralise decisions towards the teams... and the teams delegate them, one by one, to the same AI model. The organisation chart says distribution; cognition says monoculture. This chapter names the risk —cognitive re-centralisation— and proposes its countermeasures, because it is the least visible systemic risk of the whole adoption of AI: it shows up in no particular incident, but in the slow convergence of every decision towards one and the same filter.

22.1 The paradox: distribute the signature, centralise the filter

The decentralisation of block 2 pursued two dividends: speed (authority next to the information) and diversity of judgement (many different orientations looking at reality from different points). Mass delegation to a single model preserves the first and dissolves the second: each team decides locally, but all consult the same oracle, with the same biases, the same training data and the same "orientation" in the sense of chapter 13. The result is decisions that are formally distributed and cognitively homogeneous.



Formally distributed teams, all consulting the same model: the signature is shared out, the filter is centralised.

What is insidious is that the convergence feels good: less friction between teams, coherent recommendations, easy consensus. But it is the harmony of the monoculture chapter 13 already warned about: a single blind spot shared by the whole organisation, invisible precisely because nobody is outside it to point it out. Diversity of judgement was a property of the system —nobody had designed it, it came free with human diversity— and the properties that come free are the ones lost without anybody signing off their removal.

22.2 Countermeasures

The protection of cognitive diversity is designed, like everything in this guide. Diversity of sources and models: for decisions that matter, check more than one model, or the same model with deliberately different orientations; uniformity of vendor is efficiency in the trivial and fragility in the

critical. Structured dissent: human red teams and rotating devil's advocate roles that keep the muscle of disagreeing alive, complementing —not delegating— the one in chapter 17. And making the model's orientation explicit: treating the AI's recommendation as one more input with known authorship and bias ("this is what a model trained this way would say") instead of as the neutral voice of the data; naming the filter is half of neutralising it.

The alarm signal. When all the proposals from different teams in an organisation start to look alike —same structure, same options, same things ruled out—, before celebrating it as alignment it is worth asking how much of that coherence is convergence of judgement and how much is one and the same model answering everybody. The test is cheap: ask two teams for the same decision without AI and compare the variance.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain cognitive re-centralisation and why it can coexist with impeccable formal decentralisation.
- Detect signals of homogenisation of judgement: convergence of proposals, identical things ruled out, consensus without deliberation.
- Apply the countermeasures: diversity of sources and models in critical decisions, structured dissent and making the model's orientation explicit.
- Assess the balance between the efficiency of vendor uniformity and the cognitive fragility it introduces.

Common mistakes and antipatterns

Celebrating convergence as alignment. Reading homogeneity of proposals as organisational maturity. Alignment is built with shared intent (chapter 4); homogeneity of oracle merely resembles it.

Diversity of models as bureaucracy. Requiring a check against several models for every decision, trivial ones included. Like every safeguard in this guide, it is proportional: it is reserved for the irreversible and the strategic.

The ceremonial red team. Appointing devil's advocates who disagree with nothing. Structured dissent needs the same conditions as chapter 18: time, competence and dissent at no cost.

Treating the model's output as neutral data. Presenting the AI's recommendation without authorship or bias, as if it were "what the data say". Every orientation has a filter; hiding it is centralising without audit.

Further reading

· DDI — Hot Leadership Topics 2026

BLOCK 8 · DESIGNING THE HYBRID HUMAN–AI ARCHITECTURE

Chapter 23. Preserving and training judgement · EMERGING

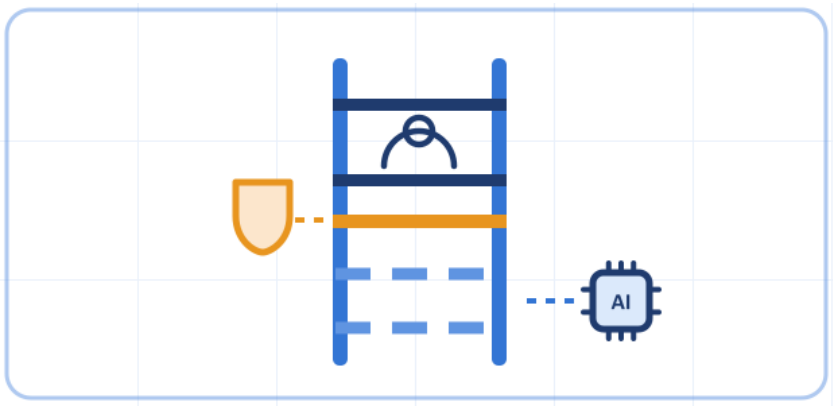
Judgement is trained by deciding small things and living with the consequences: if AI absorbs the small decisions, where will the judgement for the big ones come from?

Introduction

The guide ends where the system's future is at stake. Everything above assumes that competent human judgement exists to orient, decide, oversee and audit. But judgement is not a natural resource: it is cultivated, and its age-old method of cultivation —deciding small things and living with the consequences— is exactly what AI is absorbing. The question of this chapter is the long-term debt of the whole architecture: where will tomorrow's judgement come from if AI makes the decisions it used to be trained on today?

23.1 The severed ladder

Professional judgement was always built by climbing a ladder: minor decisions, cheap mistakes, growing judgement, major decisions. AI is cutting away the lower rungs: small decisions —which library, how to structure this module, what to reply to this customer— are precisely the ones that pay best to delegate, because they are frequent, reversible and the agent handles them well. The calculation is impeccable in the short term and leaves a gap in the long: the next generation of professionals reaches the big decisions without having paid the tuition of the small ones.



AI cuts away the lower rungs of the ladder of judgement; protected decisions rebuild them.

The risk is acute in junior profiles, whose ladder starts at the rungs that are disappearing, but it is not limited to them: senior judgement also atrophies without exercise. The 2026 trend analyses converge on the diagnosis: with scale and speed solved by AI, the system's constraint is human judgement —the precision of the questions, the depth of the interpretation, the ability to turn what has been generated into good decisions— and cultivating it stops being a by-product of the work and becomes a design decision.

23.2 Protected decisions: the training cost of judgement

The central practice is counter-intuitive and therefore needs a name: protecting small decisions so that people make them, even though AI could make them better. It is not inefficiency: it is the training cost of judgement, budgeted on purpose. The exact analogy comes from aviation: pilots still

fly by hand periodically even though the autopilot is statistically superior, because the day the autopilot is not enough a pilot who knows how to fly will be needed — and that competence is only maintained by exercising it.

The implementation uses the pieces of the guide: in the hybrid map of chapter 21, the human lane includes a deliberate quota of delegable decisions that are not delegated, marked as protected and assigned with a formative intent; decision pathways grade each person's exposure to decisions of increasing weight, as chapter 3 grades the transfer of authority; and the decision retrospective of chapter 11 acts as the gym: that is where the process is reviewed, the biases are neutralised and the learning that previously came only with the accumulation of years is accelerated.

The map's long-term investment. Protected decisions are to judgement what tests are to code: a present cost that buys future capacity. Without them, blocks 6 and 7 are left without raw material within a generation: there will be no competent supervisors for the proposal lane and no judgement to put where the whole guide says it has to be put. It is the only practice in this guide whose return is not visible within the quarter — and the one that decides whether the rest still works five years from now.

What you should be able to do

Having mastered this chapter, a professional is able to:

- Explain the mechanism of the severed ladder and why optimal short-term delegation can decapitalise judgement in the long term.
- Mark protected decisions with a formative intent on the hybrid map, justifying their cost as the training of judgement.
- Design decision pathways that grade each person's exposure —especially junior profiles— to decisions of increasing weight.
- Use the decision retrospective as an accelerator of decision-making learning, compensating for the loss of natural practice.

Common mistakes and antipatterns

Optimising each decision separately. Delegating everything the agent does better, decision by decision. Each delegation is rational; the sum decapitalises judgement. Judgement is a property of the system, and it is managed at the level of the system.

Protected decisions without consequences. Reserving decisions for people but cushioning all of their mistakes. Judgement is trained by living with consequences; the protection is of the opportunity to decide, not of the outcome.

Training in tools only. Responding to the atrophy of judgement with AI courses. Fluency with the tool does not replace the judgement exercised by deciding; they are different competences and the second is not taught in slides.

Protecting out of nostalgia. Reserving for humans decisions with no formative value "because it was always done this way". Protection is an investment with a portfolio criterion: you choose the decisions that train, not the ones you are sorry to let go of.

Further reading

· IMD — 2026 AI trends for leaders

© 2026 Scrum Manager®. This work is published under a Creative Commons Attribution-NonCommercial 4.0 International licence (CC BY-NC 4.0). Official Scrum Manager trainers and centres are licensed under the terms of CC BY 4.0 for their training activity.