Business Strategy as Code: A Governed Method for Digital-Transformation Strategy

Chapter 3 — The Digital-Transformation Chain
September 2026
Aliaksei Ivanouski

Abstract

A pillar admitted from Chapter 2 names a coded, directional change; it does not yet say which operating model delivers it, in what sequence, or against what target. Business Strategy as Code (BSaC_) — the Digital-Transformation Chain is a method that turns the admitted pillar set into a governed, metricated, sequenced digital-transformation strategy, never a delivery plan, running a forced sequence of ten Links, `L1` to `L10`, across four acts. Diagnose reads digital maturity against the pillars and forces a falsifiable vision from their domains. Bridge & grid turns that vision into an operating model changed on all five dimensions at once, then grids the data, AI, technology and governance-design enablers that make it real. Complete sequences the enablers into a staged, value-cased roadmap with named revision conditions and one falsifiable metric target. Red-team & bound designs the governance that funds and adjusts the portfolio, and marks where this chapter's authority ends and Chapter 4's begins; a pillar that fails entry, or whose admitted assumption is later falsified, returns to Chapter 2, never repaired in-chain. This paper specifies Chapter 3 and marks every rule by level. It claims no more than that: a digital-transformation strategy whose vision sentence and pillar assumptions are traceable and falsifiable. No study shows this chain produces better strategic decisions, or higher transformation success rates, than an unstructured or differently structured approach.

About this document

- **Language.** English.
- **Illustrations.** Generic and informative: no company names, and no fixed count beyond the method's own — four acts, ten Links.
- **Nature.** A method description for consultancy and advisory work, not legal, financial or investment advice.
- **Status.** Published 22 September 2026; last updated 28 September 2026.
- **Companion guide.** A practitioner guide with detailed procedures, worksheets and a full worked case is in preparation.
- **Relationship to Chapters 2 and 4.** The entry contract (§3) admits only an already-published Chapter 2 pillar, never re-opening, re-scoring or re-deriving it, and refers a failed or later-falsified one back to Chapter 2's stop 15. This chapter defines a governed, sequenced digital-transformation strategy and stops at that boundary; mobilising, scheduling, budgeting and running it are Chapter 4's own acts.

1. Introduction

1.1 The problem and the use case

An admitted pillar closes one question and leaves three open. It names a coded, directional change, but supplies no operating-model choice, no build order for the enablers that change implies, and no definition of when the change would count as achieved. An organisation can hold a set of sound pillars, a budget and a mandate, and still spend a transformation initiative's first year buying tools onto an operating model nobody changed — digitisation wearing transformation's name. The distinction a pillar does not settle on its own is exactly the one this chapter forces: whether a change reaches the operating model itself, or only speeds up the model that was already there.

Digital transformation, on this chapter's own governing claim, is not a parallel workstream running alongside a strategy a business already has. It is that strategy's operating-model consequence, traced pillar by pillar, never invented free-hand. This chapter's ten Links exist to close exactly that gap: each takes what the Link before it produced and adds one further, real choice — which operating model, which enablers, in what sequence, what counts as success — handing forward an artefact bound by a stated check, never a free composition.

This chapter serves whoever must turn an admitted pillar set into something an organisation can fund, sequence and govern: transformation leads, technology and operations leaders choosing an operating model, and the governance forums that will fund or halt what this chapter stages. It fits the same decisions Chapters 1 and 2 fit — costly to reverse, facing outside scrutiny, or changing hands. It does not replace judgement in choosing an operating model; it records how that choice was derived from the pillars it inherits, so that someone other than its author can check it.

1.2 Conventions

Requirement-level key words (MUST, SHOULD, MAY) follow the shared conventions at whitepaper.strategyascode.org/conventions/. Here they are fixed independently of Chapters 1 and 2, and a MUST binds the digital-transformation strategy.

Chapter 2 fixes "gate" as one of its eight quality gates and "stop" as one of its fifteen numbered units; Chapter 1 fixes "transform" as its own four units. This chapter reuses none of the three for its own things. It calls its own ten operations **Links**, written bare — `L1` … `L10` — never with a middle dot, which stays Chapter 2's code-family notation. Each Link admits its input under its own **admission**

check and releases its output under its own **contract check**, in exactly the sense the earlier chapters already publish. A single **entry contract**, given in full in §3, governs which of Chapter 2's pillars this chain may run against. It issues one of two verdicts, `ADMITTED` or `REFERRED` — the second returning an unfit pillar to Chapter 2's own stop 15 rather than repairing it here. What the chain produces, once every check has passed, is **a digital-transformation strategy** — the name used throughout this paper for Chapter 3's output. It is never called "a plan," which names only Chapter 4's later, separate act of mobilising, scheduling and running what this chapter has defined.

This chapter also keeps Chapter 1's word **challenge** for its own adversarial test set; §7.1 states, once, how Chapter 3's own challenges relate to Chapter 1's.

1.3 How to read this paper

§3 gives the entry contract; §4 gives the chain as four acts; §5 gives all ten Links, plus the exit contract (§5.11) and the eight challenges (§5.12). §6 covers evidence and provenance; §7 the failure modes and non-claims; §8–§10 applicability, alternatives, scope and references. Appendix A and B hold the rule table and the derivation register.

1.4 Abbreviations

This chapter mints its own Link and artefact codes (§4); the short forms below are the abbreviations its prose otherwise uses.

- AI — Artificial Intelligence
- API — Application Programming Interface
- BSaC_ — Business Strategy as Code
- COBIT — Control Objectives for Information and Related Technologies
- CTO — Chief Technology Officer
- MACH — Microservices-based, API-first, Cloud-native and Headless
- MBA — Master of Business Administration
- MIT CISR — MIT Center for Information Systems Research
- RFC — Request for Comments
- TOGAF — The Open Group Architecture Framework
- TOM — Target Operating Model

2. Foundations

Chapter 2 draws its foundations from requirements traceability, architecture decision records, evidence-based management and Rumelt's kernel; Chapter 1 from problem-formulation and confirmation-bias literature. This chapter draws on three different bodies instead, none restated from the earlier chapters: Capgemini Consulting and the MIT Center for Digital Business's (2012) two-axis digital-maturity read, feeding L1; Rogers's (2016) five domains of change, feeding L2's mapping surface; and Weill and Woerner's (2018) two-dimensional digital-business-model framing, behind L2's orientation fork. None of the three supplies the vision itself — only the admitted pillars do that (§5) — and this chapter borrows each body's shape, never its scored thresholds or its specific findings about which reading, domain or orientation outperforms another; those claims sit outside this paper's own non-claims (§7). The foundations bound how L1 and L2 read the pillars, never what they may contain. §9 gives the fuller account of what each body contributes and what is deliberately not borrowed from it.

3. The principle

3.1 The rule at the pillar seam

A pillar can pass Chapter 2's own field discipline — a coded parent, a resolvable source — and still arrive here too thin to run a chain against. Chapter 2 states: "Pillars are the sinks of the derivation graph…" A sink is where Chapter 2's own checking stops, not where Chapter 3's begins. The rule that carried every stop of the analysis chain — no output without a traceable input — holds at this seam too, though what it tests changes shape: nothing arriving here is a raw claim awaiting evidence, but an already-admitted artefact awaiting a different question — not whether the pillar is right, but whether it is shaped enough to enter this chain at all.

Chapter 3 inherits the admitted pillar set, `P·1 … P·n`, and nothing else; the pillars are fixed inputs this chain never re-opens or re-scores, by design, not oversight. Chapter 3 carries no authority to repair one, and no chain to run without one that clears a minimum shape first. A pillar is a choice, not yet a transformation, and the entry contract tests only the second half of that sentence: not whether the choice itself was right, only whether enough of it survived Chapter 2's own coding to give this chain something to read.

3.2 The entry contract

This chapter's entry contract is built the way Chapter 1 built its own two-sided check: an admission check facing what arrives, a contract check facing what leaves. Every field it tests — a parent code, a source, a scope — is already a Chapter 2 field; it only asks whether enough of them are present, and pointed the right way, to start.

A pillar is admissible only where four conditions all hold. It MUST name a coded, directional change in capability or offering, never an aspiration. It MUST trace to at least one Chapter 2 parent code — a TOWS option, an ERRC action, a 7S gap, or, where the pillar carries one, a VRIO verdict or an opportunity/threat code. Its parent chain MUST resolve to a named external source or a registered

internal fact, reusing Chapter 2's own graph-root rule rather than inventing a second one. It MUST state what would be observably different if it were achieved, extending to this seam the same consequence test Chapter 1 applies to a problem statement. And it MUST be scoped to an identifiable organisational domain or unit — narrow enough to seed a maturity read and a domain mapping in the two Links that follow immediately.

The verdict vocabulary is deliberately narrower than Chapter 1's own: two verdicts only. **ADMITTED** applies where all four conditions hold and the chain runs. **REFERRED** applies where one or more fails. A referral MUST name the clause that failed and return the pillar to the analysis chain's own stop 15 for correction. REFERRED is deliberately never written as Chapter 1's DEFICIENT: a diagnosis Chapter 1 calls deficient is repairable inside Chapter 1's own transforms, but a pillar Chapter 3 refers is not repairable inside Chapter 3 at all.

Admission is not a single pass a pillar clears and leaves behind. A tracked metric, or a re-run of the chain's own falsifiability test, can later show an admitted pillar's underlying assumption no longer holds. When that happens, Chapter 3 MUST issue a second, formal referral to Chapter 2's stop 15, naming the falsifying evidence and the pillar concerned. Chapter 3 MUST NOT patch the pillar, the vision, or the operating model in-chain to absorb the new evidence. Doing so would re-derive a pillar outside Chapter 2's own scoring discipline — exactly what this chapter already has no authority to do.

This is not Chapter 4's `GC3` stage-gate, which governs whether a wave's funding continues, tested on tracked movement, and never asks whether a pillar's own assumption still holds.

The chain MUST proceed on whatever pillars are currently admitted; a pillar still REFERRED does not halt it. Once later admitted, only the Links whose codes cite it are re-checked.

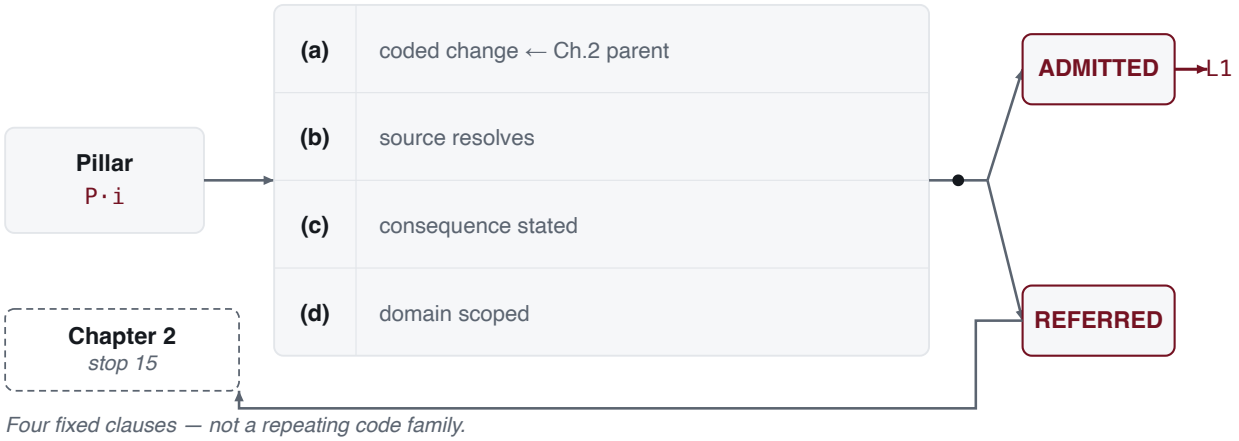


Figure 3.1 — The entry contract: admitting a pillar. A pillar is admitted only once all four clauses hold; failing any one refers it back to Chapter 2, never repaired in-chain. This method's format.

3.3 Two-sided contracts

Chapter 2's own stops need only a contract check, since each takes as its input the already-checked output of the stop before it. The entry contract is where Chapter 3 breaks that pattern once, on purpose: a pillar is the first artefact this chain receives that no Chapter 3 check has ever touched. That two-sided model — an admission check facing what arrives, a contract check facing what leaves — is the one Chapter 1 built for exactly this situation; Chapter 3 applies it here at full strength, at the seam, rather than distributing it Link by Link.

Past that seam, the chain behaves like Chapter 2's own, not like Chapter 1's: each Link takes as its input an already-checked upstream artefact (L2 takes L1's capability read alongside the admitted pillars, L3 takes L2's coded vision, and so on to the exit contract), already cleared by either the entry contract or an earlier Link's own contract check — so no Link needs a second admission check, only its own contract check on what it produces.

The one exception is the post-admission referral (§3.2 above): a pillar the entry contract has already admitted can still later fail, never through a missed contract check but because the fact underneath it changed. That failure surfaces downstream and travels back through the single connection the entry contract already opened, never through a new one; only the entry contract, and its post-admission trigger, look back past the chain to the pillar itself.

3.4 What is and is not the contribution

None of this section's material is new — an admission check, a narrow verdict vocabulary, a referral naming its own failed clause are all established practice in some form. What Chapter 3 contributes is holding that practice to the same two-sided discipline Chapter 1 established and Chapter 2 already runs, at a seam neither chapter's own contract reaches. The contribution is auditability, and nothing wider than that; it stops well short of an outcome claim (§9 gives the fuller argument, and the entry contract's own checkable, non-outcome scope).

4. The chain

4.1 Four acts at a glance

The chain runs as four acts across its ten Links: Diagnose, Bridge & grid, Complete, and Red-team & bound. Each act closes before the next opens, and nothing in a later act substitutes for a check an earlier one already performed.

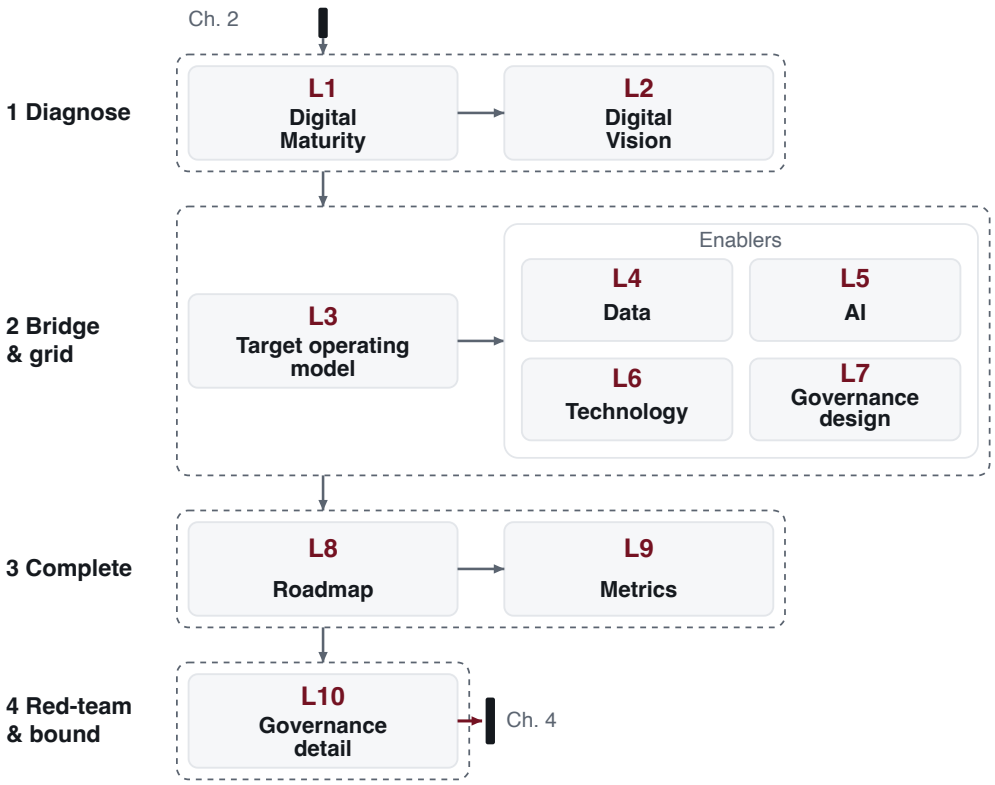


Figure 4.1 — The digital-transformation chain: four acts, ten links. Read top to bottom: ten links run in strict sequence through four stacked acts, from the admitted pillars entering above Diagnose to the digital-transformation strategy handed to Chapter 4 below Red-team & bound; the bridge & grid act holds the target operating model beside one grid block holding its four enabler links. This method's format.

Diagnose (Links 1 and 2) reads the admitted pillars against where the organisation actually stands today, then forces — never brainstorms — a falsifiable statement of where it points. Digital Maturity comes first because a vision written before an honest capability read tends to describe an aspiration rather than a position; Digital Vision comes second because it is the first Link whose output every later Link, directly or indirectly, cites. Neither Link asks how large the transformation should be — both ask only how honest the starting point is and how testable the direction chosen from it turns out to be.

Bridge & grid (Links 3 through 7) is where a stated direction either becomes a change to how the organisation actually runs, or stays a slide with a deadline attached. The Target Operating Model turns the vision into a five-dimension model of people, process, data, technology and governance, reached by way of a capability map, never guessed directly from the vision sentence. The four enabler Links then grid what makes that model real: data, artificial intelligence, technology and governance design, each a forced decision, one per cell, never a catalogue of purchases. This act carries the chain's two strongest rules, one at the bridge and one inside the grid — the act in which a transformation is most easily mistaken for a technology rollout.

Complete (Links 8 and 9) takes the grid's decisions and stages them into waves, bounded by whatever resourcing window the organisation actually has, then attaches a value case whose targets are numeric, dated and owned rather than aspirational. Red-team & bound (Link 10) closes the chain: it designs, but does not run, the cadence that will hold every wave to account, and draws — explicitly, in the chain's own exit contract — the line at which this chapter's own work ends and the next chapter's begins.

A pillar enters this chain only once it has cleared the entry contract Chapter 2 hands it through (§3); the chain does not run against pillars that contract has referred. What leaves the far end, once Link 10 closes, is named once and consistently: a digital-transformation strategy, never a delivery plan (§6).

4.2 Why this order

The order is not a preference. Each act needs what the one before it already produced, and skipping ahead manufactures an output nobody asked the chain to certify.

Digital Vision cannot force a falsifiable sentence without Digital Maturity's honest capability read behind it — an orientation chosen before anyone has looked at what the organisation can actually do is a guess wearing a vision's clothes. The Target Operating Model cannot be reached directly from a vision sentence either: the chain runs vision to capability gaps to operating model, deliberately never vision straight to an organisation chart, since a model assembled without first naming the capabilities a vision demands tends to rearrange boxes rather than change what any of them decide. The enabler grid depends on the operating model the same way — a data, technology or governance decision taken before the model exists to use it is a purchase in search of a justification.

Inside the grid itself, one dependency is load-bearing enough to carry its own rule. An artificial-intelligence decision MUST NOT be taken before the data decision it depends on has produced a working source of truth. An AI use case reasoning over data nobody has yet made trustworthy is exactly the technology-first failure the grid exists to prevent (§5). The same logic governs Complete at coarser grain: a roadmap cannot stage initiatives it has not yet derived from the grid. Enablers MUST reach working readiness before any use case that depends on them is allowed to scale.

Red-team & bound runs last because testing a portfolio's governance and boundary against a chain that does not yet exist tests nothing. Two mergers were considered and rejected in this design — folding the enabler grid into the operating model Link, and folding governance design into the roadmap — both of which would have hidden a distinct decision inside another Link's card, costing a reader the ability to see and challenge each decision on its own terms. The sequence is reasoned from what each Link needs already to be true, not benchmarked against an alternative ordering: no source in this paper's reference register compares chain orderings against each other on results.

4.3 The chain at a glance

The table below gives the shape of all ten Links side by side. §5 gives every one of them in full, as a card, in the same six-field format.

Link	Act	Takes	Produces	Contract check
L1 Digital Maturity	Diagnose	The admitted pillars' gap side	Capability read, leadership read, verdict, scope line	Verdict names both axis readings and their pillar sources
L2 Digital Vision	Diagnose	The capability read; the admitted pillars	Orientation; five domain reads; the coded vision sentence	Vision sentence is falsifiable; every clause traces to a named domain code
L3 Target Operating Model	Bridge & grid	The vision sentence	Capability map; five-dimension operating model	Every capability cites a parent; all five dimensions populated or the gap named
L4 Data	Bridge & grid	The operating model	The data-enabler decision	One decision recorded, never a list
L5 AI	Bridge & grid	The operating model; the data-enabler decision	The AI-enabler decision	Names its gating dependency on the data enabler; never ready ahead of it
L6 Technology	Bridge & grid	The operating model	The technology-enabler decision	One decision recorded, never a list
L7 Governance-design	Bridge & grid	The operating model	The governance-enabler decision	One decision recorded, never a list
L8 Roadmap	Complete	The enabler grid	Placed candidates; staged waves	Every candidate cites a parent, tracing unbroken to an enabler; waves respect the sequencing law
L9 Metrics	Complete	The roadmap	North-star and supporting metrics	Each metric cites its pillar or enabler parent
L10 Governance detail & synthesis	Red-team & bound	The governance-enabler decision; the roadmap	Decision rights; funding model; review cadence; the strategy pack; the boundary marker	All three governance fields present; the design/run boundary carried in spirit

Read left to right, a row states what a Link needs before it acts and what its output must show once it has. Read top to bottom, one Link's Produces column is very often the next row's Takes column — the dependency §4.2 defends, laid flat. No cell states a value; every field names a code family or an artefact, never a number. Where a Contract check column names more than one condition, both are MUST-level, and neither substitutes for the other.

5. The ten links

All ten Links are this chain's own. What follows gives all ten Links in full, as cards, each with exactly six fields in the same fixed order — purpose, inputs, outputs, derivation rule, contract check, anti-pattern — closed by one generic illustration.

5.1 L1 — Digital Maturity

RECEIVES		HANDS ON
Admitted pillars' gap side entry contract (§3) →	L1 **Digital Maturity** Diagnose	→ **Capability read, leadership read** to L2 **Verdict, scope line** scope line carries no further derivation

CONTRACT CHECK · MUST — The verdict MUST name both axis readings and the pillar evidence each rests on.

Purpose. Digital Maturity opens the chain by reading the admitted pillars against where the organisation genuinely stands today, before any vision work names where it should point. It exists to catch, early and cheaply, the confusion the whole first act works to prevent: mistaking a technology purchase for a change to how the organisation runs. Its output colours every Link that follows, without blocking any of them outright. A low reading is not a reason to halt, only a reason to be honest about how far the operating model still has to move. A high reading carries the opposite risk: it can make a later Link's claims read as more proven than the placement itself actually supports.

Inputs. The admitted pillars' gap side — the capability they name as missing or underused, carried forward from the entry contract (§3).

Outputs. Four outputs: the capability read and the leadership read, placed on a two-axis grid; the verdict, stating where that placement lands; and the scope line. The scope line is held apart from the other three, because nothing in the chain derives from it in turn.

Derivation rule. The placement borrows its two-axis shape from Capgemini Consulting and the MIT Center for Digital Business's 2012 study of digital leaders. That study plots an organisation's digital intensity against its transformation-management intensity, across four named quadrants — Digirati, Fashionistas, Conservatives and Beginners. This Link reuses the quadrants as diagnostic labels, never as a ranking to optimise against.

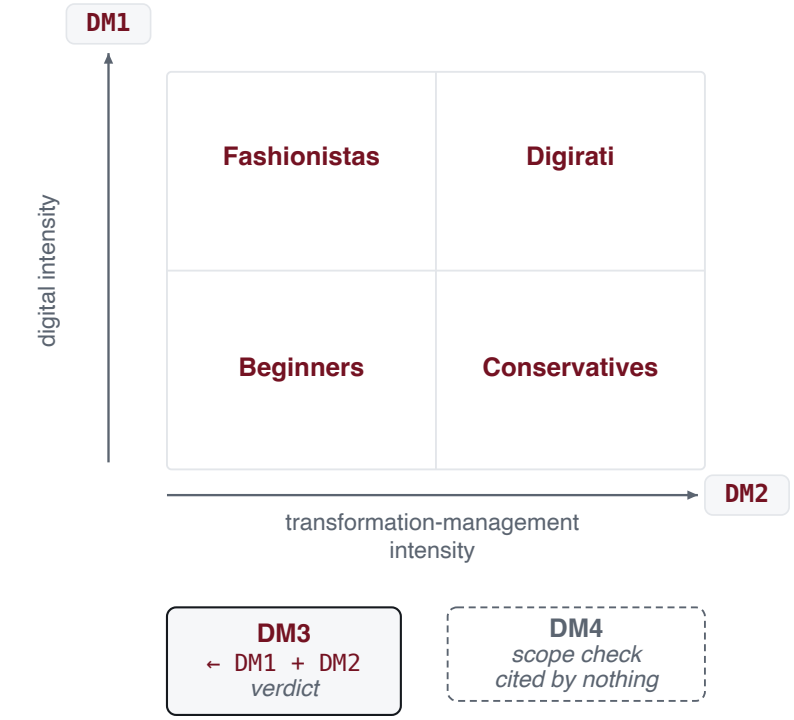
The placement is diagnostic colour, not a pass/fail bar. No numeric threshold in the literature says how much of either axis is "enough," and this chapter invents none. Reviews of maturity models generally document the same limitation (Andersen et al., 2020; Becker et al., 2009): a static, retrospective read, not a tested account of how an organisation actually progresses.

The verdict, `DM3`, is a placement: it cites both axis readings and, on the pillar evidence behind them, states where the organisation currently lands — arguable on that evidence, not automatic. The scope line, `DM4`, is a different kind of statement: a check on what the admitted pillars would amount to if the operating model never changed, since technology bought onto an unchanged model is digitisation, not transformation. It is recorded but cited by nothing, sitting deliberately outside the chain's derivation graph rather than left out by oversight.

Contract check. The verdict MUST name both axis readings and the pillar evidence each rests on.

Anti-pattern. No anti-pattern is named for this Link. The buzzword-vision failure this chain guards hardest against belongs one Link on, at Digital Vision (§5.2).

Illustration (informative). A regional insurer's placement reads low digital intensity, moderate transformation-management intensity: real assets sit idle behind no operating model built to use them.



Dashed = cited by nothing. No numeric threshold; no position plotted.

Figure 5.1 — The maturity read. L1 plots the two-axis grid — digital intensity against transformation-management intensity — into a verdict, `DM3 ← DM1 + DM2`, which cites both readings; a separate scope line, `DM4`, carries no derivation parent and feeds nothing downstream. Neither axis carries a numeric threshold and no organisation's position is plotted; the grid states a shape, not a score. After Capgemini Consulting & MIT Center for Digital Business (2012), adapted.

5.2 L2 — Digital Vision

RECEIVES		HANDS ON
Capability read from L1 **Admitted pillars** → entry contract (§3)	L2 **Digital Vision** Diagnose	→ **Orientation** to L3 **Five domain reads** to L3 **Coded vision sentence** to L3

CONTRACT CHECK · MUST — The vision sentence MUST be falsifiable and every clause MUST trace to a named domain code; it MUST NOT derive from a domain the chain excludes from parenthood.

Purpose. Digital Vision converts a capability read into a single falsifiable sentence. The three steps that produce it SHOULD be forced deliberately, never brainstormed. A brainstormed vision is where this chapter's most expensive early failure lives: a form of words fluent enough to survive a board meeting and specific enough to decide nothing. Everything the bridge and the grid do next takes its shape from the sentence this Link is required to produce.

Inputs. The capability read from Digital Maturity; the admitted pillars themselves.

Outputs. An orientation, OR-E or OR-S, chosen and stated as leading, with the alternative kept secondary rather than dropped; five domain reads; and the coded vision sentence itself.

Derivation rule. The first step forces an orientation choice between two ways of leading a transformation: OR-E, deepening the customer's own experience of the offering, or OR-S, embedding digital capability inside the offering itself. OR-S is named Digitised Solutions — this method's own label, not MIT CISR's. The admitted pillars decide which orientation leads, never preference. This fork is this method's own convention for forcing a choice; it is not a finding that the two orientations are mutually exclusive. Both orientations are named whichever way the choice falls. The one kept secondary is still a domain this Link's contract check must be able to trace, not a discarded alternative.

The second step maps the organisation onto five domains of change, DV1 … DV5 — customers, competition, data, innovation and value — drawn from Rogers's *Digital Transformation Playbook*. Every domain SHOULD be named, even where a domain reads as a named gap. Five domains all read as assets, with no stated gap, is the same failure in different clothes, only discovered later and at greater cost.

The third step composes the sentence itself. It derives from the chosen orientation and exactly three of the five domain reads, never from all five and never from the two domains the chain holds back from parenthood. Widening the derivation to whichever domains happen to read well is how a falsifiable sentence quietly becomes an unfalsifiable one.

Contract check. The vision sentence MUST be falsifiable and every clause MUST trace to a named domain code; it MUST NOT derive from a domain the chain excludes from parenthood.

Anti-pattern. *d-VISION — the buzzword vision.* A form of words that reads as direction while deciding nothing, guarded against by the second step's naming requirement and the third step's tracing requirement together.

Illustration (informative). A logistics operator's dominant pillars force an engagement-led orientation; its innovation domain is named amber, not dropped, since experimentation is not yet a built capability.

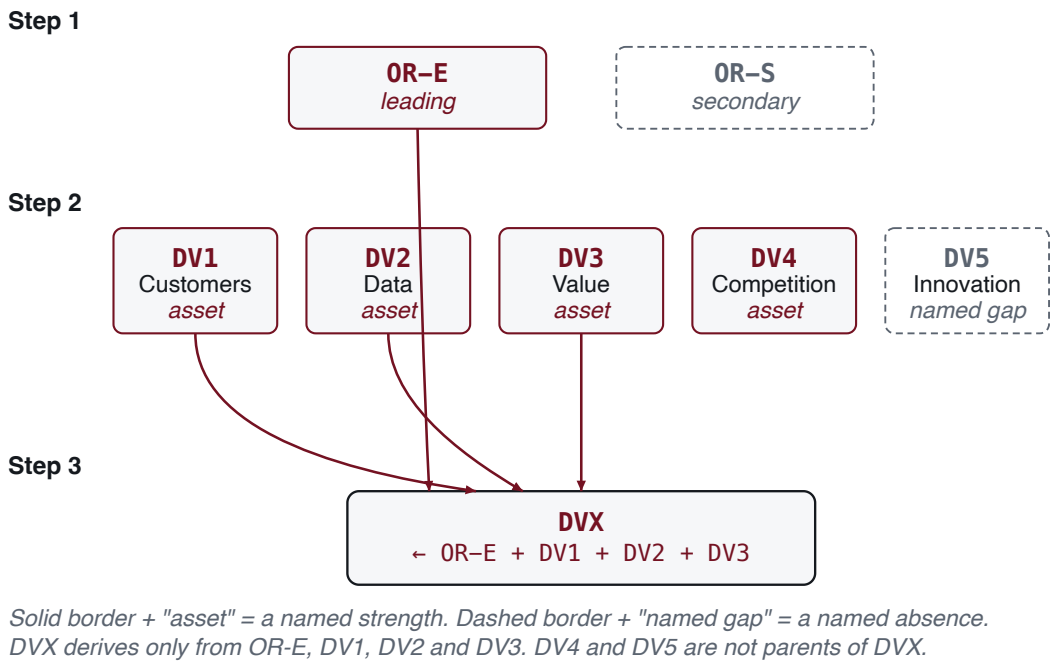


Figure 5.2 — The vision, in three forced steps. Step one forces an orientation, `OR–E` or `OR–S`, one leading and the other kept secondary, never dropped. Step two maps five domains at equal spacing, each named as an asset or a named gap. Step three composes `DVX` from the orientation and exactly three of the five domain reads; `DV4` and `DV5` send no edge and are never its parents. This method's format.

5.3 L3 — Target Operating Model

RECEIVES		HANDS ON
Vision sentence from L2 →	L3 **Target Operating Model** Bridge & grid	→ **Capability map** to L4, L5, L6, L7 **Five-dimension operating model** to L4, L5, L6, L7

CONTRACT CHECK · MUST — Every capability MUST cite a domain or pillar parent; all five operating-model dimensions MUST be populated, or the gap named explicitly.

Purpose. The Target Operating Model is the pivotal bridge of the whole chain. It is the Link at which a stated direction either becomes a change to how the organisation actually runs, or quietly turns back into a technology purchase wearing a transformation's name. Everything Diagnose produced was preparation for this one Link; everything the grid produces afterwards depends on what it decides here. No other Link in the chain carries as much weight on so few sentences.

Inputs. The coded vision sentence.

Outputs. A capability map, CAP·1 … CAP·n, naming what the vision requires; a five-dimension operating model — OM-P, OM-R, OM-D, OM-T, OM-G — spanning people, process, data, technology and governance respectively.

Derivation rule. The bridge runs vision to capability gaps to operating model, never vision directly to an organisation chart. Capabilities are named first because they state what the vision requires; the model is only how the organisation staffs and runs them.

The chapter's strongest rule sits here: the operating model MUST change on all five dimensions at once. A model that changes technology while leaving people, process, data and governance exactly as they were is digitalisation, not transformation, however large the technology line item.

Where the model's people and governance dimensions call for outcome-accountable teams rather than order-taking ones, that pattern draws on Cagan and the Silicon Valley Product Group's product-operating-model literature. The governance-design Link (§5.7) reuses the same literature for a different field of the same model; this method states the source once, here, rather than arguing it twice.

Contract check. Every capability MUST cite a domain or pillar parent, with none invented; all five operating-model dimensions MUST be populated, or the gap in a dimension MUST be named explicitly rather than left blank.

Anti-pattern. *d-TOM — no operating-model change.* A document can carry a data workstream, a technology workstream and an artificial-intelligence workstream, and still never touch who decides, in what process, with which people. This is this Link's own failure mode. It is not closed here, but at the chain's red-team act, once the whole portfolio exists to test.

Illustration (informative). A retailer's capability map names a returns capability and an inventory-visibility capability; its model populates process and data fully, names a people gap explicitly, and defers governance to later in the grid.

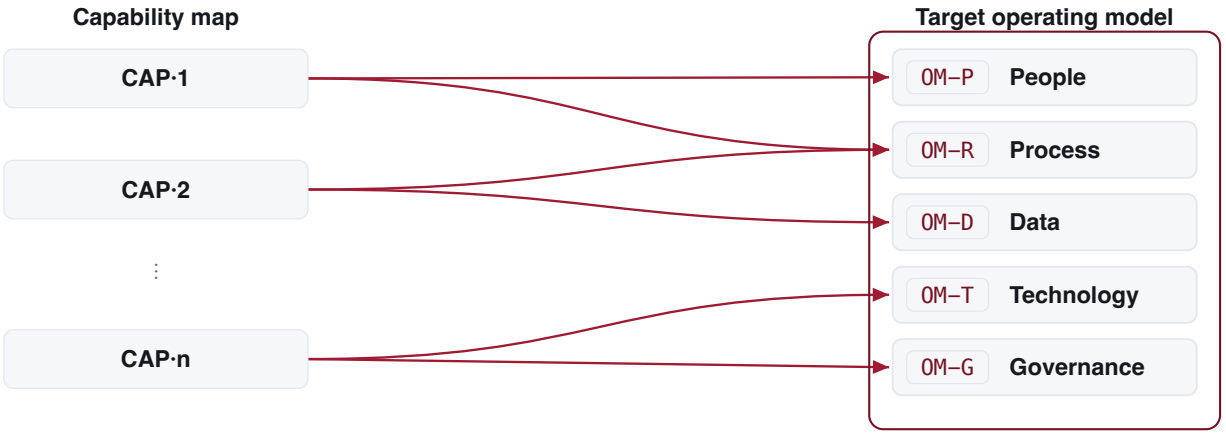


Figure 5.3 — Capability map to target operating model: where the codes sit. Every capability code cites the `OM` dimension it changes, and all five dimensions must carry at least one such citation before the operating model counts as changed. This method's format.

5.4 L4 — Data

RECEIVES	L4	HANDS ON
Operating model → from L3	**Data** Bridge & grid	→ **Data-enabler decision (EN-D)** to L5, L8

CONTRACT CHECK · MUST — The data enabler MUST record one decision, never a shopping list of platforms; a defensive requirement is stated as a floor, not the brief.

Purpose. Data is the first of four enabler decisions that make the operating model real. Each enabler cell answers exactly one question rather than accumulating a shopping list. Data's question is asked first because the chapter's second-strongest rule — artificial intelligence waits on data — makes every other enabler's timing depend on what gets decided here. Whatever the operating model's data dimension named as a gap, this Link is where the organisation commits to closing it in a specific direction. That direction, not a technology stack, is what the next three enabler cells inherit.

Inputs. The operating model, specifically its data dimension.

Outputs. The data-enabler decision, EN-D: a single named choice, not a portfolio of options.

Derivation rule. The decision follows the offence-versus-defence framing DalleMule and Davenport set out in *Harvard Business Review* (2017). It points the data enabler at revenue and insight first, holding the defensive work — portability, governance of the data itself — to whatever the offensive use case actually demands.

This method then architects the enabler around one authoritative source of truth, rather than several partial, competing versions. That design choice sits on top of the offence/defence priority; the article itself does not specify it.

Offence leads because the value case this chain states later, at Metrics, lives on this side of the choice. A data enabler that leads with defence still has to answer, eventually, what the offensive case was — and answering it late costs the wave that follows its readiness date.

Contract check. The data enabler MUST record one decision, never a shopping list of platforms; a defensive requirement is stated as a floor, not the brief.

Anti-pattern. *d-TECH — technology-first,* in this cell's own form: a data platform purchased ahead of a named offensive use case, before the operating model that would use it has been decided. This is the anti-pattern the whole enabler grid exists to prevent. Buying the platform does not retroactively supply the decision it was meant to serve; the next card extends the same failure to artificial intelligence specifically.

Illustration (informative). A telecom reseller's data enabler commits to one customer-behaviour source of truth, treating required payment-data lineage as the floor, not the starting brief.

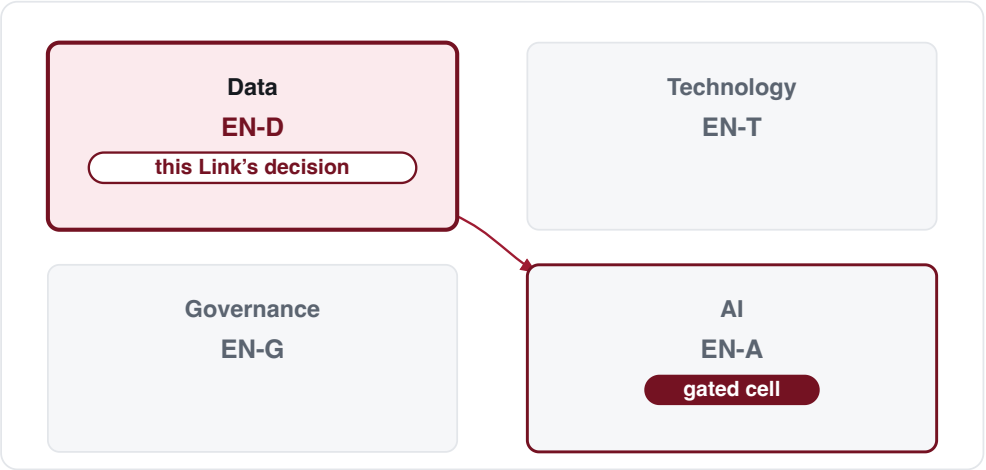


Figure 5.4a — The enabler grid: the data decision. The grid highlights the data cell, EN–D; the other three cells sit muted for context. The AI cell stays gated on this decision until it is made. This method's format.

5.5 L5 — AI

RECEIVES		HANDS ON
Operating model from L3 **Data-enabler decision** from L4 →	L5 **AI** Bridge & grid	→ **AI-enabler decision (EN-A)** to L8

CONTRACT CHECK · MUST — The AI enabler MUST name its gating dependency on the data enabler explicitly; it MUST NOT be recorded as ready before that dependency is.

Purpose. Artificial Intelligence is the second enabler decision, and the one Link in the grid this chain explicitly holds back. It exists to state, and enforce, that an AI use case is downstream of a working data source of truth — never a parallel initiative that happens to start on the same day. It is deliberately the narrowest of the four enabler cells: breadth here is exactly how the grid's own shopping-list failure re-enters through the back door, this time wearing a model instead of a platform contract.

Inputs. The operating model; the data enabler's own decision and its readiness.

Outputs. The AI-enabler decision, EN-A: a narrow portfolio of use cases, each named against the data dependency that constrains it.

Derivation rule. The sequencing law is this chapter's second-strongest rule: an AI initiative MUST NOT start before the data enabler's source of truth exists. An AI use case reasoning over data nobody has yet made trustworthy is the technology-first failure in its purest form.

No standards-grade or peer-reviewed source defines a numeric threshold for when data "exists" in the sense this rule needs. This method states that gap rather than inventing a figure to close it.

The closest analogue in the literature treats AI readiness as a multi-dimensional institutional question — governance, capability, compliance, trust — not a single number. Its own evidence base is public-sector deployment, not enterprise transformation; it is named here only as an analogy, never as this Link's own authority.

Where the organisation genuinely cannot yet say whether its own source of truth is ready, the honest answer is to say so and hold the enabler back. Substituting a plausible-sounding date for a test nobody has actually run is not an honest answer.

Contract check. The AI enabler MUST name its gating dependency on the data enabler explicitly; it MUST NOT be recorded as ready before that dependency is.

Anti-pattern. *d-TECH — technology-first,* in its AI-specific form: an AI initiative committed, staffed or purchased before the data enabler it depends on has produced a working source of truth. This is the same failure Data guards against above, recurring one cell over.

Illustration (informative). A distributor holds a demand-forecasting model back until its inventory source of truth is live, rather than building it on a parallel track.

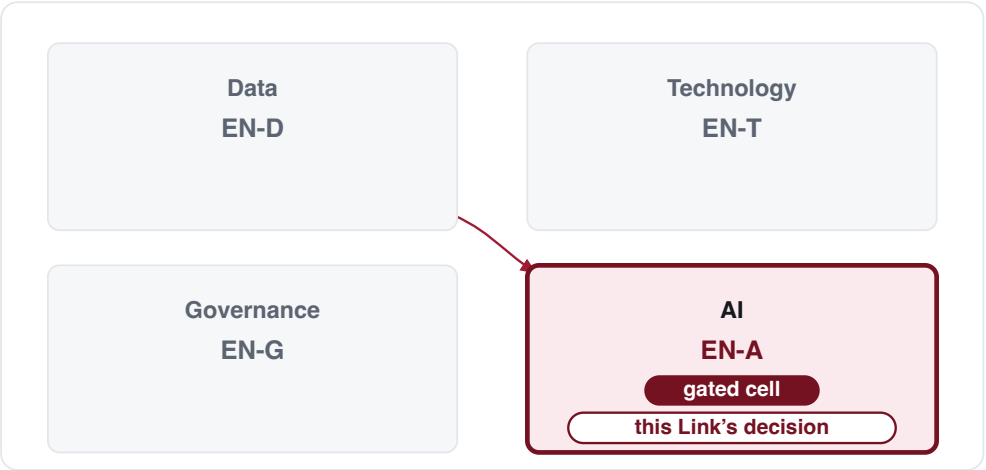


Figure 5.4b — The enabler grid: the AI decision. The grid highlights the AI cell, EN–A, gated on the data decision beside it; the other three cells sit muted for context. This method's format.

5.6 L6 — Technology

RECEIVES		HANDS ON
Operating model from L3 →	L6 **Technology** Bridge & grid	→ **Technology-enabler decision (EN-T)** to L8

CONTRACT CHECK · MUST — EN-T MUST cite OM-T or a named capability-map parent, and MUST record exactly one decision, never a vendor list or a menu of options under evaluation.

Purpose. Technology is the third of the grid's four enabler decisions, and the one most easily mistaken for the whole transformation. A vendor demonstration is the easiest thing in the grid to confuse with progress. It produces something visible before any of the other three cells have had to commit to anything at all. L6 exists to keep that visibility from substituting for a decision. It makes the grid ask what the operating model requires of its technology, rather than what technology happens to be on offer this quarter.

Inputs. The operating model, specifically its technology dimension, OM–T; the capability map's technology-facing gap.

Outputs. The technology-enabler decision, EN–T: a single named choice, not a shortlist.

Derivation rule. The decision follows a composable, API-first pattern over a single monolith platform, rebuilding only the specific integration surfaces the operating model's own gap names — never "modernise the stack" stated as a standing intention.

The pattern's own source is an industry consortium, the MACH Alliance, not a peer-reviewed literature. MACH — Microservices-based, API-first, Cloud-native and Headless — names the composable architecture style L6 derives towards. This method states that plainly, rather than dressing an industry convention as a finding; its basis is accordingly lighter than the chain's own cited academic sources.

The enabler grid's own shared rule still binds L6 exactly as it binds the other three cells: each cell MUST record one decision, never a shopping list. A purchase order naming no capability gap and no OM dimension has decided nothing, whatever it cost.

Contract check. EN–T MUST cite OM–T or a named capability-map parent, and MUST record exactly one decision, never a vendor list or a menu of "options under evaluation."

Anti-pattern. No anti-pattern is minted specifically for L6. The grid's own d-TECH — technology chosen before the operating model demands it — is guarded across all four cells. It is sharpest here, where a vendor demonstration is easiest to mistake for a decision already made.

Illustration (informative). A regional insurer's technology cell rebuilds two claims-system connectors on public APIs, naming the operating-model dimension each one serves. A third, older core system with no public API is named as a known gap, rather than papered over with a workaround.

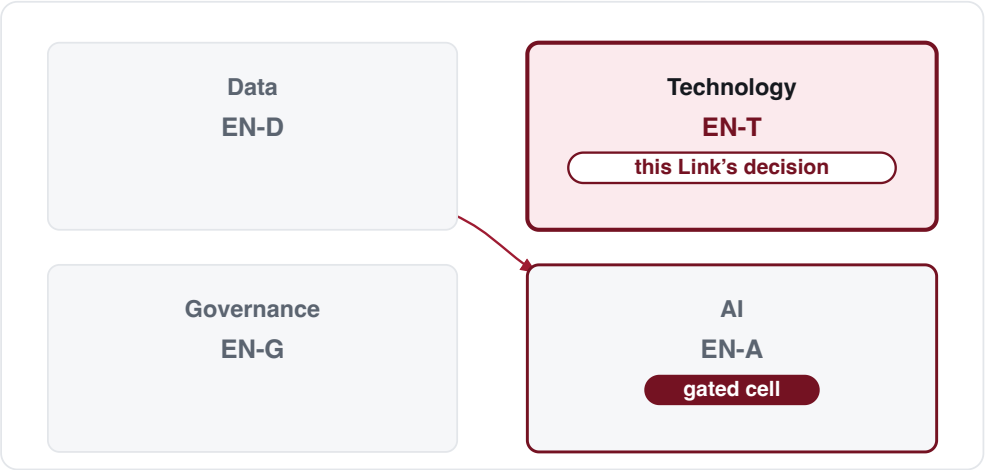


Figure 5.4c — The enabler grid: the technology decision. The grid highlights the technology cell, EN–T; the other three cells sit muted for context, including the AI cell, gated on the data decision. This method's format.

5.7 L7 — Governance-design

RECEIVES		HANDS ON
Operating model from L3 →	L7 **Governance-design** Bridge & grid	→ **Governance-enabler decision (EN-G)** to L8, L10

CONTRACT CHECK · MUST — EN-G MUST cite OM-G or a named capability parent, and MUST record one decision, never a menu of governance models still under discussion.

Purpose. Governance-design closes the enabler grid with the decision every other cell has been assuming exists: who actually holds the right to decide, once the operating model's other four dimensions are built. Without it, the grid's other three decisions sit on an organisation nobody has yet said will run them.

Inputs. The operating model, specifically its governance dimension, OM–G; the capability map's governance-facing gap.

Outputs. The governance-enabler decision, EN–G: a single named choice about decision rights, not yet a cadence or a funding model.

Derivation rule. The decision is a product-operating-model design, drawn from Cagan and the Silicon Valley Product Group's account of the product operating model. Outcome-owning teams hold decisions a committee with staged sign-off would otherwise hold.

Design is the operative word. EN–G states who will decide and on what basis, not how a cadence will run. That boundary belongs to L10. L7 does not anticipate it.

What EN–G fixes is narrower than it sounds: a named owner for each outcome the capability map assigns. It is not a restructured reporting line, and not yet the funding or cadence mechanics that turn ownership into a working governance model.

A cell that names an owner but leaves the basis for their decisions unstated has produced half a decision. The missing half is exactly what L10 later checks for.

Contract check. EN–G MUST cite OM–G or a named capability parent, and MUST record one decision, never a menu of governance models still under discussion.

Anti-pattern. No anti-pattern is named specifically for L7. The grid's own d-TECH applies here as it does at L6: an off-the-shelf governance tool is not a decision about decision rights. A hollow EN–G cell instead produces decision rights drawn as an org chart, rather than as who decides what, on what data. That failure is red-teamed at L10, where the chain tests whether governance survived into the finished strategy.

Illustration (informative). A logistics operator's governance cell names two product squads as outcome owners for the initiatives their capability map touches. It also names the portfolio board as the body that still owns funding checkpoints — a decision-rights split, not an organisation chart.

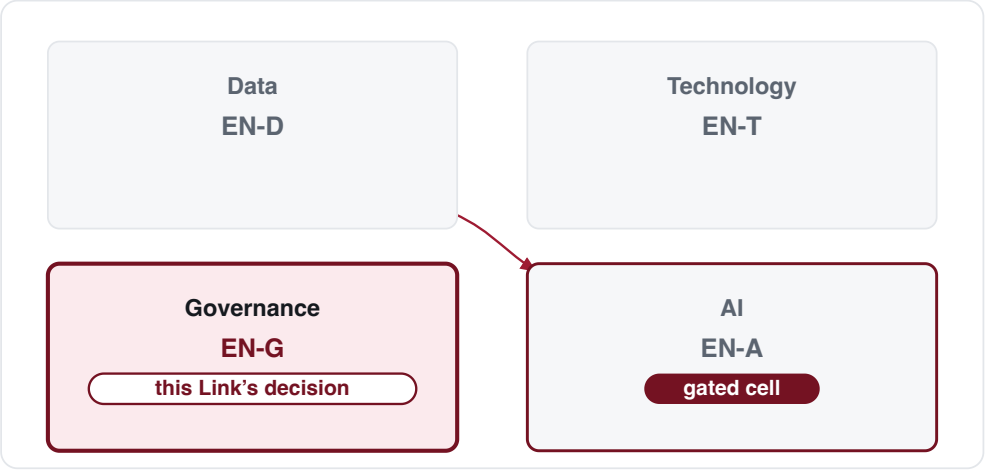


Figure 5.4d — The enabler grid: the governance decision. The grid highlights the governance cell, EN–G; the other three cells sit muted for context, including the AI cell, gated on the data decision. This method's format.

5.8 L8 — Roadmap

RECEIVES		HANDS ON
Enabler grid — data, AI, technology, governance decisions → from L4, L5, L6, L7	L8 **Roadmap** Complete	→ **Placed candidates (I1 … In)** to L9 **Staged waves (RW1–RW3)** to L9, L10

CONTRACT CHECK · MUST — Every I# MUST cite a parent — an enabler decision or a depended-on initiative — and that chain MUST reach an enabler unbroken; a candidate with no parent MUST be refused outright. Every wave MUST respect the sequencing law; each wave transition MUST name its clearing metric, threshold and tested assumption. An initiative MUST NOT be staged in the same wave as, or earlier than, any initiative it depends on — the grid only orders initiatives whose dependencies are met.

Purpose. The roadmap takes the finished grid's four decisions and turns them into a sequenced, dated commitment rather than a list of good ideas. It exists because a grid full of correct decisions still fails if every initiative it implies starts on the same day; staging, not the decisions themselves, is this Link's own job.

Inputs. The finished enabler grid: the data, AI, technology and governance decisions together.

Outputs. Candidate initiatives, `I1 … In`, each placed by value and feasibility; three waves, `RW1` Foundation, `RW2` Scale, `RW3` Differentiate, staged against a bounded clock; and, for each wave transition, a named revision condition.

Derivation rule. Placement on the value-feasibility grid is a SHOULD-level convention, not a MUST. No primary academic source establishes the 2×2 itself, and its closest lineage, Henderson's 1970 growth-share matrix, plots different axes entirely. A practitioner who defends a different placement against one named alternative has not broken the method.

The sequencing law binds far more strictly, at MUST-level. The grid ranks initiatives by value and feasibility, and dependencies then decide the wave each one enters (Figure 5.5). Enablers MUST reach readiness before their use cases scale, and an initiative dependent on the AI enabler MUST NOT be committed ahead of the data enabler it depends on. An initiative MUST NOT be staged in the same wave as, or earlier than, any initiative it depends on; the grid only orders initiatives whose dependencies are met.

Some initiatives' parent edges are method-inferred rather than individually quoted, and one AI-enabler edge records a readiness check rather than an execution. Both distinctions, and their basis, are stated in full where the chain's evidence is assembled (§6), not repeated here.

Each wave transition carries its own revision condition. This method adds it; the transition MUST name which K# metric clears before the next wave commits in detail, MUST state the threshold value itself, and MUST name the assumption the clearing tests. Naming the metric alone, with no threshold and no assumption, does not satisfy this rule.

This chain MUST define these conditions as part of L8's own artefact, and MUST NOT run them. That cadence belongs to Chapter 4's own quarterly stage-gate, which releases the next wave's funding only once the named metric moves. The threshold values themselves are the practitioner's own, supplied against their instance of the method; this chain fixes none beyond the illustration below.

Contract check. Every I# MUST cite a parent — an enabler decision or a depended-on initiative — and that parent chain MUST reach an enabler decision unbroken; every wave MUST respect the sequencing law; and each wave transition MUST name its clearing metric, threshold value and tested assumption together, never the metric alone. An initiative MUST NOT be staged in the same wave as, or earlier than, any initiative it depends on; the grid only orders initiatives whose dependencies are met. A candidate that cites no parent at all MUST be refused outright, whatever value or feasibility it claims.

Anti-pattern. d-OCEAN — attempting every initiative at once, so the foundation moves that unblock everything else receive the same starved attention as the ones still blocked behind them.

Illustration (informative). An unnamed claims-processing vendor commits RW2 only once K1 clears 80%, testing that the foundation enablers hold under live load. It commits RW3 only once K2 clears 55%, testing that the scale-wave initiatives generalise past the first cohort.

The table below places one illustrative set of initiatives against the enabler grid (informative). Its source column names the enabler decision an initiative cites directly — I1 through I3, and the two AI initiatives, I6 and I7 — and for the rest, the dependencies column carries that trace instead, which is what keeps every initiative auditable back to the grid. Appendix B carries the full register.

I#	Initiative	Value	Feasibility	Source	Depends on
I1	Telemetry data SSOT (pipeline and model)	High	High	EN–D (inferred)	—
I2	Rebuild of two core connectors on public APIs	High	High	EN–T (inferred)	—
I3	Stand up the first empowered product squad	High	Medium	EN–G (inferred)	—
I4	Self-serve onboarding path	High	Medium	—	I1, I2, I3
I5	Telemetry-to-CS health scores in the workflow	High	Medium	—	I1
I6	AI onboarding auto-mapping	Medium	Low	EN–A (inferred)	I1, I2
I7	AI stall-prediction	Medium	Low	EN–A (inferred)	I1, I5
I8	E-invoicing-ready package	Medium	Medium	—	I4

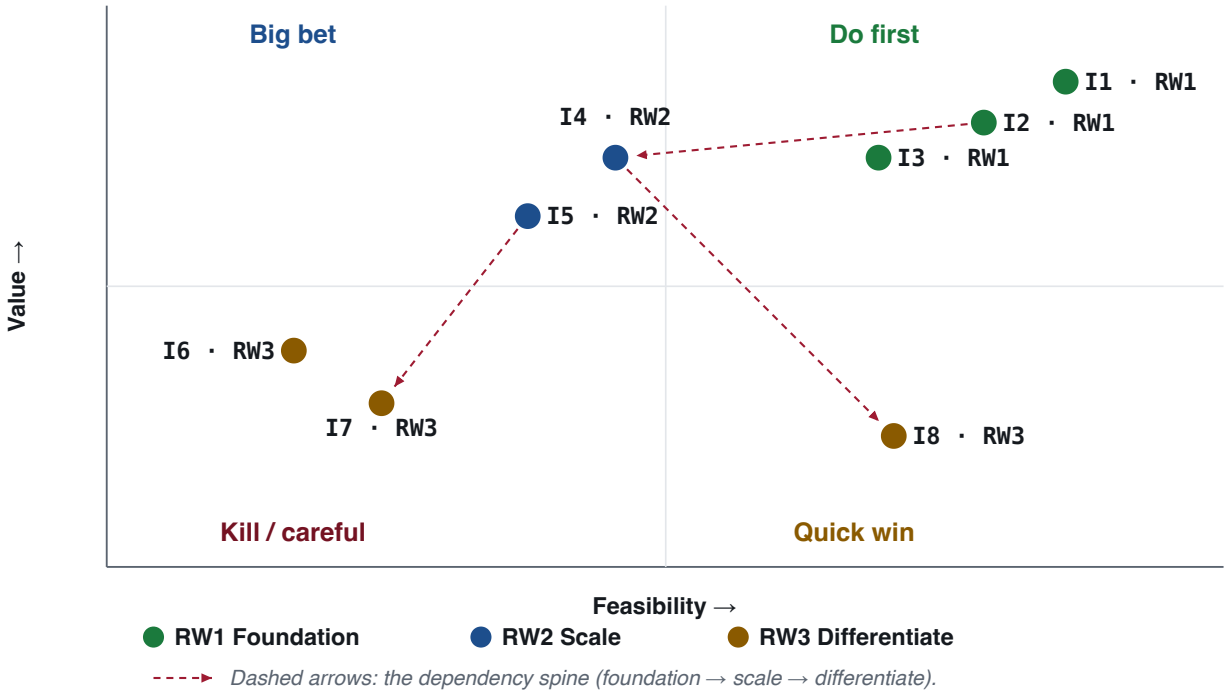


Figure 5.5 — Initiatives ranked, staged and linked (informative). Position ranks one illustrative set of initiatives by value and feasibility, and colour marks the wave each one enters. The dashed arrows trace the dependency spine — foundation, then scale, then differentiation — and the full dependency list is in Appendix B. This method's format.

Dimension		RW1 Foundation	RW2 Scale	RW3 Differentiate
OM–P	People	I3	—	—
OM–R	Process	—	I4	I8
OM–D	Data	I1	I5	I6 I7
OM–T	Technology	I2	—	—
OM–G	Governance	—	—	—

Columns are the three roadmap waves in order; a dash marks a cell no initiative lands in.

Figure 5.6 — The roadmap swimlane (informative): one illustrative placement. This shows one illustrative set of initiatives placed by wave and operating-model dimension; the method's rule is that every initiative sits in its wave and names the operating-model dimension it changes. The Data row carries `I1` in RW1 and `I6`/`I7` in RW3, illustrating the AI initiatives staged only after the data initiative that they depend on. This method's format.

5.9 L9 — Metrics

RECEIVES		HANDS ON
Staged roadmap from L8 →	L9 **Metrics** Complete	→ **North-star metric (K0)** to L10 **Supporting metrics (K1–K5)** to L10

CONTRACT CHECK · MUST — Each K# MUST cite its pillar or enabler parent, never floating free as a number someone liked.

Purpose. Metrics turns the staged roadmap into a value case a challenger can trace, built as a tree with one root rather than a dashboard of numbers nobody owns. It exists so that "the transformation is working" has one falsifiable number behind it, not a felt impression. That number also needs a stated origin, not a plausible-sounding round figure chosen after the fact.

Inputs. The staged roadmap: the waves and their initiatives.

Outputs. One North Star, `K0`; a leading layer, K1 and K2, that predicts movement in it; a lagging layer, K3 through K5, that confirms it once a wave has run.

Derivation rule. `K0` MAY be adopted as this North-Star mechanism. No established or peer-reviewed source requires it: it is a practitioner coinage associated with writing on growth metrics, not a standards-body or peer-reviewed finding. A practitioner MAY set a value-case root by another method entirely.

Every target on the tree SHOULD be numeric, with a named baseline, a target and a date. This is an analogical extension of Locke and Latham's goal-setting research, stated as an extension, not as that research's own finding — Locke and Latham studied individual task performance, not a portfolio metric tree.

A lagging metric MAY be framed as a recovery against a measured prior fall, rather than a restoration to some earlier peak. This method illustrates the pattern once, as its own worked convention, not as a generalisable rule for every K# a practitioner might set.

Contract check. Each K# MUST cite its pillar or enabler parent, so a metric on the tree traces to the artefact it is meant to move, never floating free as a number someone liked.

Anti-pattern. d-PILOT — a metric that stayed a pilot's own vanity number, never re-tied to the operating-model dimensions, `OM–P` or `OM–R`, that would prove the change survived contact with scale.

Illustration (informative). A subscriptions operator's K0 is the median days to a new account's first measurable value, with a dated baseline and a target roughly a third of it. K1 is cited to a named enabler parent, rather than left to float beside the tree.

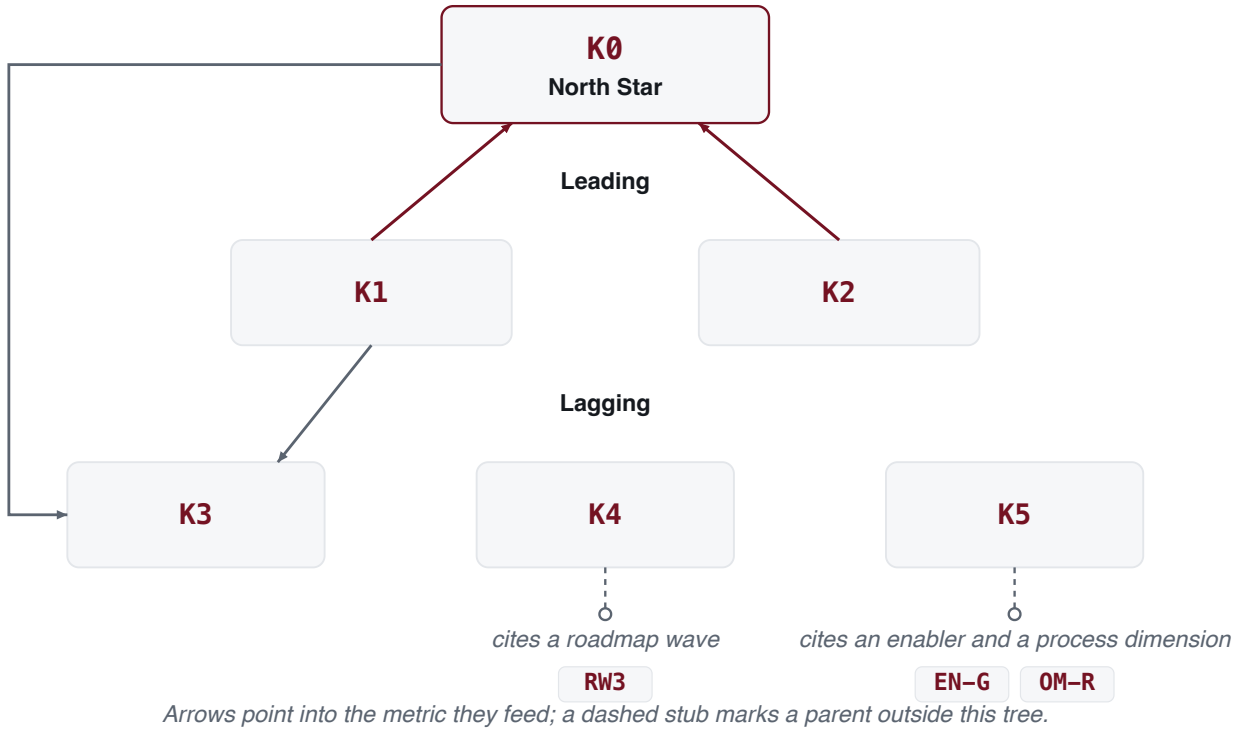


Figure 5.7 — The metric tree (informative). The tree shows one illustrative set of metrics: the North Star K0 at the root, the leading metrics K1 and K2 that drive it, and the lagging metrics K3 through K5 that confirm it once a wave has run. This method's format.

5.10 L10 — Governance detail & synthesis

RECEIVES
Governance-enabler decision
from L7
Staged roadmap with metric tree
from L8, L9

→

L10
Governance detail & synthesis
Red-team & bound

→

HANDS ON
Decision rights, funding model, review cadence
GOV1–GOV3
Strategy pack
HX1
Boundary marker
HX2 — names what Chapter 4 owns

CONTRACT CHECK · MUST — GOV1, GOV2 and GOV3 MUST all be present. GOV2's funding model MUST release each wave's budget only against K0 movement. The design/run boundary MUST be carried in spirit wherever GOV3 is stated.

Purpose. Governance detail & synthesis closes the chain. It turns EN–G's single decision into a working governance model. This is where the chain tests, for the last time, whether the operating model it built at L3 actually survived everything built on top of it. That test spans the roadmap's sequencing, the metric tree's targets, and, finally, who decides what happens when either one moves.

Inputs. EN–G, the governance-enabler decision; the staged roadmap with its metric tree.

Outputs. GOV1, decision rights; GOV2, a funding model; GOV3, a review cadence (Figure 5.8) — together with the finished strategy pack, HX1, and the boundary marker, HX2, that hands it to Chapter 4.

Derivation rule. GOV1 names which product squads own which outcomes, and what a portfolio board still holds.

GOV2 derives from two inputs: RW1's wave structure and K0's baseline. It MUST be a funding model that releases each wave's budget only against metric-tree movement. The same funding model also watches K3 as a longer-horizon target, never as a derivation parent it cites.

GOV3 is defined inside this chain, never run. Its "defined" bar carries one addition this method adds: a completeness checklist modelled on the Evaluate-Direct-Monitor split ISACA's COBIT draws between governance and management. GOV3 SHOULD name all four fields — cadence, inputs, decision rights and escalation. This is a SHOULD-level addition, not a MUST. It is stated plainly as this paper's own elaboration; no primary source fixes the four fields, only the presence of GOV3 itself.

Decision rights of this shape draw in part on Weill and Ross's account of IT-governance archetypes — who decides, not only what is decided. This method carries that account into a product-operating-model setting, rather than the enterprise-wide one it was written for.

Contract check. GOV1, GOV2 and GOV3 MUST all be present. GOV2's funding model MUST release each wave's budget only against K0 movement, with K3 tracked as the target that movement watches rather than a co-equal release condition. The design/run boundary MUST be carried in spirit wherever GOV3 is stated. This chain draws the governance; Chapter 4 operates it.

Anti-pattern. d-TOM, finally closed rather than merely named. A strategy can carry a technology, data and AI workstream and still fail here, if no decision right, no funding checkpoint and no cadence changed alongside them. L10 tests whether OM-P, OM-R and OM-G are still visible in GOV1 through GOV3. It also tests whether governance quietly reverted to whoever held the budget before the grid was built.

Illustration (informative). A mid-market retailer's GOV2 ties its funding checkpoint to K0 and to the quarter K3 is first expected to move. Its GOV3 names the quarterly cadence's inputs and escalation path without naming who chairs it — a decision Chapter 4 makes once the cadence runs.

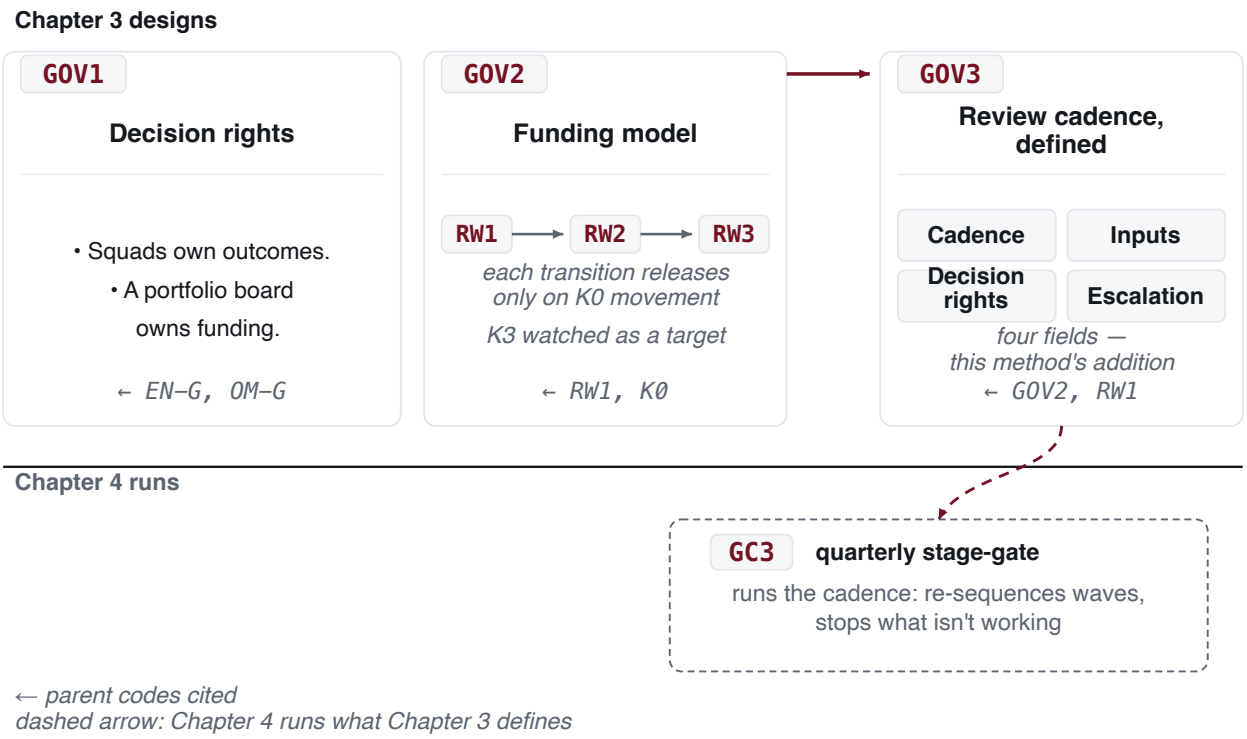


Figure 5.8 — Governance, designed not run. GOV1 fixes decision rights, GOV2 fixes a funding model that releases each wave's budget only against K0 movement, and GOV3 defines a quarterly review cadence; Chapter 3 designs all three, and Chapter 4's GC3 stage-gate is what runs the cadence. This method's format.

With these three artefacts complete, the chain's output is whole. HX1 MUST be named, here and in every figure caption that shows it, as a digital-transformation strategy, never a plan. What that pack contains in full, and the boundary it crosses into Chapter 4, is given next, immediately below; the chain's evidence for how it was derived is assembled at §6.

5.11 The exit contract: from the pack to the boundary

The chain's output is a single consolidated pack, and it earns a code of its own. It carries five components:

- the vision and its orientation;
- the target operating model across all five dimensions, with the capability map (CAP·1 … CAP·n) beneath it;
- the prioritised and sequenced portfolio, with its three waves and every dependency intact;
- the governance design, across decision rights, funding checkpoints and review cadence;
- the metric tree, with its North Star.

Nothing in the pack carries a live number in this paper — every field stays structural — but nothing is missing either. A pack short one of the five components is not yet the chain's output, whatever else it contains. Naming all five in one place, every time the pack is described, lets a later reader check completeness at a glance. It saves that reader from hunting through ten separate Link cards to confirm nothing was quietly dropped between them.

A second code marks the boundary the pack crosses, and this is the sharpest line the chain draws. What leaves the room is the pack itself — a shaped, sequenced, governed digital-transformation strategy, ready to be acted on and deliberately not yet acted on. What does not leave the room belongs instead to Chapter 4's delivery plan. That plan carries the detailed scheduling and budgeting it requires, the risk register, the change-management mechanics that turn a governed strategy into adoption, and the running of the review cadence this chain only designed.

Chapter 3 draws the governance; Chapter 4 operates it. Chapter 3 states the North Star and the tree beneath it; Chapter 4 tracks them in flight. None of that downstream work is this chain's to perform. The boundary code MUST name each item Chapter 4 owns explicitly, rather than leaving a reader to guess where the fence sits.

The naming matters as much as the boundary itself. What crosses MUST be named a digital-transformation strategy everywhere this paper states it — the pack, the boundary marker, this figure's own caption below — never a plan of any kind. A pack can carry a workstream for every dimension the chain's own principle requires, sequenced against a stated clock and wired to metrics that trace back to a P·i. Such a pack is a strategy ready for Chapter 4 to take into delivery. Forming it is this chain's own act; running it is Chapter 4's. The two are never named as the same act.

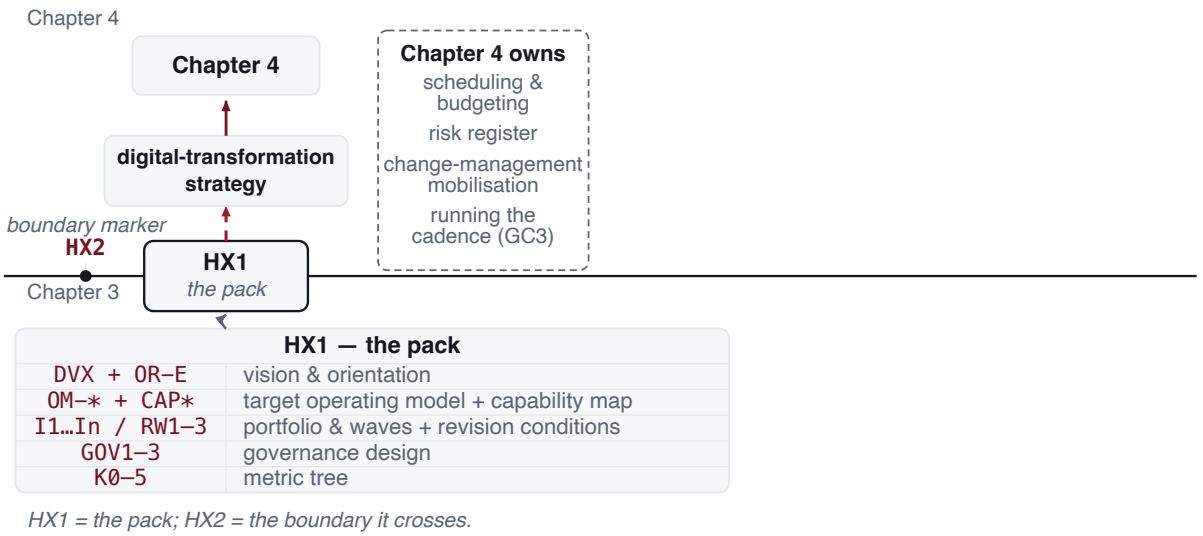


Figure 5.9 — The exit contract: from the pack to the boundary. HX1 consolidates the chain's five components into one pack; HX2 marks the line it crosses, naming what Chapter 4 owns and what stays behind. This method's format.

5.12 The eight challenges

The eight challenges attack a Link's output at the chain's own recurring points of failure — vision, model, sequencing, prioritisation, measurement, governance, boundary, orientation — never from a generic checklist. Numbering follows the fixed order Chapter 3 publishes and does not change.

1. Falsifiability. "What would make DVX false, and have you looked for it?" Pass condition, MUST: a named, obtainable observation exists that would show the vision false. The search for that observation MUST actually have been run, not merely imagined. A vision immune to any future observation has not passed the three forced steps that define it; it has been asserted in their shape. Failing this challenge fails L2 outright — DVX MUST be recomposed, not patched, because a vision that cannot be shown false carries nothing for the rest of the chain to trace against. This is the chapter's own analogue of Chapter 1's falsification challenge, run here against a vision rather than a statement.

2. Five-dimension completeness. "Name the one OM dimension you did not change." Pass condition, MUST: all five operating-model dimensions are populated, or the gap is named and defended as a deliberate deferral, not an oversight. A bridge that fills in technology and data while leaving people, process or governance silent has not crossed from vision to model; it has swapped one slogan for a more technical one. Failing this challenge fails L3, and the diagnosis it forces has a name: d-TOM, the chapter's own climactic failure, arriving early because a bridge left silent on a dimension is where it starts.

3. Sequencing and dependency. "Show me the data readiness this initiative depends on, before I look at the initiative itself." Pass condition, MUST: any enabler decision that carries a gating dependency on another names it, evidenced, not assumed. An AI use case proposed before the data it needs is ready has skipped a step the grid itself defines. This holds however finished the initiative otherwise looks. Failing this challenge fails L5, or the wave of L8's roadmap the initiative was placed in. A dependency left unevidenced is not a detail to fix later — it is the initiative in the wrong wave. The question runs backwards, from initiative to precondition, because that is the direction a reviewer checks and a rushed roadmap skips.

4. Value-and-feasibility defensibility. "Why is this initiative placed there, and what is the alternative ranking?" Pass condition, SHOULD: the placement is defended against at least one named alternative ranking, or changed. Value and feasibility scores have no primary-literature source behind their placement rule (§7.5); the discipline this challenge enforces is not that a score is provably right,

but that it was argued, not assumed. Failing this challenge does not fail the roadmap outright — the L8 initiative is re-placed, and the argument that moved it is recorded alongside the one it replaced. A later reader can then see the placement was contested and settled, not merely asserted twice.

5. Metric fitness. "Is the North Star the right measure of the thing you say matters?" Pass condition, SHOULD: the headline metric is defended against at least one named alternative measure, or changed. A metric tree can be internally consistent — every leading number feeding the number above it — while measuring the wrong thing at the root, and internal consistency alone cannot catch that. Failing this challenge does not fail the whole value case; the contested L9 metric is changed, and the case for the replacement is recorded. This challenge is deliberately narrower than falsifiability: it asks only whether the North Star is the right number to watch, not whether it could be shown wrong.

6. Governance completeness. "Cadence, inputs, decision rights, escalation — which of the four does the governance design leave unnamed?" Pass condition, MUST: all four fields are named, not a decision-rights statement standing in for the whole design. A governance design that names who decides but not how often, on what inputs, or what happens when a decision is contested has drawn half a boundary and called it complete. Failing this challenge fails L10. A design missing one of the four fields is not a lighter version of a complete one — it is a gap a later reviewer cannot distinguish from an oversight. Nothing in the artefact says which it was.

7. Design/run boundary. "Point to the sentence that says this is Chapter 4's to run, not Chapter 3's." Pass condition, MUST: the handoff names its own boundary explicitly. It states what leaves the room designed, and what Chapter 4 owns from here. A pack can hand over a governance design, a staged roadmap and a value case without ever stating that running them is somebody else's act. Doing so leaves the seam this chapter sits on unmarked. Failing this challenge leaves the HX1 pack incomplete, whatever else it contains, because an unmarked boundary is exactly the condition under which a governed design quietly turns into an unaccountable one.

8. Orientation-fork defensibility. "Is your lead orientation really exclusive of the other, or could both be pursued at once?" Pass condition, SHOULD: the fork is defended as this method's own convention for forcing a choice. It is never cited as a finding that the two orientations cannot coexist. The literature behind the fork does not establish the two as mutually exclusive (§2); what the chain enforces is a forced choice of emphasis, not a proven trade-off. Failing this challenge re-argues L2's orientation, and the pillars that broke the tie the first time are checked again against the alternative before the chain proceeds.

6. Evidence, provenance, the handoff

6.1 What the derivation graph shows

A chain earns its audit trail the same way the admitted pillars did: nothing enters it uncited. Every code the chain mints names its parent — the pillar it traces to (P·i), or the earlier code it was derived from. This runs from the first maturity reading through to the governed, sequenced portfolio, and a challenger can walk any code's citation back to an admitted pillar without a gap anywhere on the way.

One code is a deliberate exception. L1's scope line carries no parent citation, by L1's own design (§5.1), and not by an oversight in the graph. As the paragraph below states, it also feeds nothing downstream. Every other code in the chain satisfies the citation claim above without exception.

Figure 6.1 draws this as a schematic: one node per code family per band, linked family to family rather than code to code. The schematic draws only the main family-to-family flow; further long-range citations — a metric that cites a pillar directly, for example — are listed in Appendix B instead. The full instance sits in Appendix B, built against four admitted pillars (informative: the count is the instance's, not the method's). That instance holds fifty-two nodes — the four pillars plus forty-eight codes the chain mints across its diagnosis, its operating-model bridge, its enabler grid, its portfolio, its metric tree and its governance design. One hundred and eight of its edges are stated; five more are drawn as inferred (§6.3) — one hundred and thirteen edges in total, for this instance. Neither the schematic nor the register carries a filled example, company data, or a title written inside a node.

Four nodes reach a genuine dead end: none has a path forward that reaches the pack this chain hands off (§5.11). The leadership-capability reading feeds the maturity verdict built from it, but neither the reading nor the verdict is cited by anything further. The definitional line that separates genuine transformation from mere digitisation, and the orientation this chain keeps secondary rather than leading with, are each cited by nothing at all. No path from any of the four ever reaches the exit contract; Appendix B marks each one.

The register keeps all four on record, marked as terminating, rather than dropping them. A diagnosis a later Link does not reuse still belongs in the record. Erasing it would make the chain look tidier than it actually ran — a worse failure than an honestly recorded dead branch. A register that hid its dead ends would look complete right up until an auditor asked where a named reading went. Marking the branch terminating, not missing, is what keeps that question answerable.

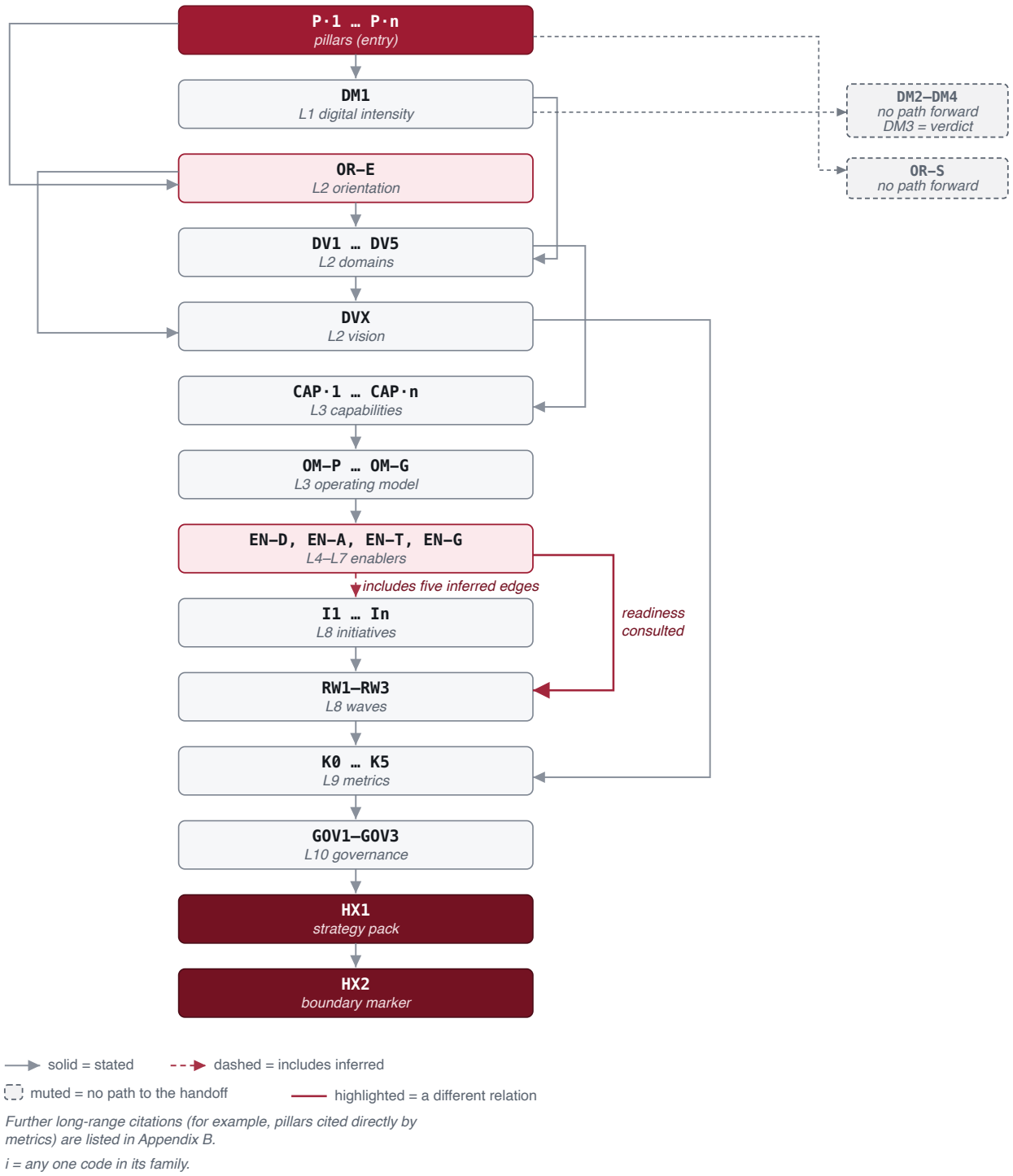


Figure 6.1 — The derivation graph, schematic (code families only; the full register is in Appendix B). Codes flow family to family, from the pillars through diagnosis, the bridge and the enabler grid to the roadmap, the metric tree and governance. A family with no path forward to the handoff stays on record, never dropped. This method's format.

6.2 Reading a derivation like a stack trace

A trace reads like a stack trace: start at any conclusion and follow each code's cited parent down, one hop at a time, until the chain reaches an admitted pillar. Two traces from Figure 6.1's instance show the shape below. A break anywhere along the way — a code with no cited parent, or a citation that resolves to nothing further back — is a finding, not a formatting slip. These codes belong to the illustrative instance Appendix B registers, not to the method itself, which fixes no code count.

Figure 6.2 — Reading a derivation like a stack trace: two traces from a conclusion back to a pillar (informative). This method's format.

Trace 1, from the Scale wave. `RW2` ← `EN–A` ← `EN–D` ← `OM–D` ← `CAP·2` ← `DV2` ← `P·2` — L8 waves ← L5 AI ← L4 data ← L3 operating model ← L3 capability ← L2 domain ← pillar.

Trace 2, from the North Star. `K0` ← `DVX` ← `DV1` ← `P·1` — L9 ← L2 vision ← L2 domain ← pillar.

6.3 Five edges, inferred rather than quoted

Most of the graph's edges are citation chips the chain states outright: a code names its parent in the same sentence that mints it. Five edges are not stated that way, and the difference is worth saying plainly, rather than smoothing over. Three are unstated: from the data, technology and governance enabler decisions to the first three foundation initiatives respectively. The other two are unstated the same way, one wave later: from the AI enabler decision to each of the two AI initiatives. The chain states the general derivation claim that its foundation-wave initiatives fall out of the enabler grid as a set, and gives every initiative its own dependency column, but does not write the individual enabler-to-initiative chip for these five the way it writes every other edge on the page.

The five are drawn anyway, on two different bases. Three rest on a content match a reader can check directly: the data initiative builds exactly the single source of truth the data decision calls for, the technology initiative rebuilds exactly the connectors the technology decision names, and the governance initiative stands up exactly the empowered structure the governance decision describes. The other two — the onboarding auto-mapping and stall-prediction initiatives — rest on availability rather than a named content match: both are AI use cases, and this chain admits an AI use case only through the AI decision, so both descend from it even though no sentence states either edge outright.

The graph marks this plainly at each of the five edges, rather than treating them as equal in kind to the edges the chain states outright — three from the general claim plus a content match, two from the general claim plus the AI decision's exclusive route — never individually cited the way the rest of the graph is. An auditor reading the graph should be able to tell an inferred edge from a stated one at a glance.

Each of the five inferred edges MUST carry its own one-sentence inference-basis clause, stated at the edge itself — the five given above — never left for the general claim alone to justify a specific pair.

6.4 One dependency, two different relations

One pairing in the graph looks, at first glance, like the same edge drawn twice. It is not. Collapsing it into a single edge would hide something the chain itself keeps distinct.

The AI enabler decision feeds the scale wave, and the same decision also feeds the differentiate wave two stages later: one source code, two destinations. The first edge records that the scale wave's own staging consults a readiness check on the AI enabler before it proceeds, even though no AI initiative actually executes inside the scale wave itself. The check is made, not yet cleared for use. The second edge records where the AI-dependent initiatives actually run, once the data foundation they are still blocked behind has had a full wave to mature. The first relation is a consultation; the second is an execution.

A merged edge would say only that the AI decision and two later waves are connected. It would lose the reason two initiatives wait a full wave before they start — the entire sequencing law the enabler grid exists to enforce. The graph keeps both edges, and the card carrying them states the distinction once, rather than leaving an auditor to infer it from the drawing alone.

6.5 The exit contract: from the pack to the boundary

The exit contract now sits at §5.11, immediately after `L10`, the Link it closes.

7. Challenges, failure modes and non-claims

7.1 The bar

A digital-transformation strategy can satisfy every clause the entry contract publishes and still choose the wrong operating model. The entry contract's four clauses test a pillar's shape, not whether the choices the chain then makes from it are the right ones. A target operating model can trace every dimension to a domain code and still be technology dressed as transformation. The bar this section sets is therefore adversarial, not procedural: a Link's output is admitted only once it has been put in front of someone equipped to break it, and has survived.

The eight challenges (§5.12), fixed as this chapter stands, are numbered separately from Chapter 1's nine. The shared shape — a fixed, numbered adversarial test set with pass conditions and named verdicts — reflects one shared design decision across the paper. It is never a shared roster, and never a constraint either chapter places on the other. Every challenge MUST be run and recorded, whatever the eventual verdict; an unanswered challenge MUST NOT be recorded as withstood. A strategy that has withstood all eight is challengeable, not thereby correct: it has been put in front of people equipped to break it, and did not break.

7.2 Who may challenge

Competence to challenge is fixed by what a Link's output claims, not by seniority. A target operating model that asserts a gap on one dimension is challenged most sharply by whoever answers for that dimension in practice, not by whoever holds the most senior chair in the room. A governance design that names a decision right is challenged, in addition, by whoever would sit in the seat it names. This mirrors Chapter 1's own challenger model at Chapter 3's own cruxes — data readiness, sequencing, metric fitness, governance completeness. It does not restate Chapter 1's own specific accountability rule for a diagnosis's baseline metric, which is Chapter 1's own construction.

The two rules the chain holds to are set at the level of the whole set, not of any one challenger's seat. Every challenge MUST be run and recorded, whatever the eventual verdict. An unanswered challenge MUST NOT be recorded as withstood. The record exists precisely so a reader who was not in the room can tell a challenge that was tested apart from one that was merely present on the list.

7.3 The eight challenges

The eight challenges now sit at §5.12, immediately after the exit contract they test against every Link's output.

7.4 Failure modes

Each Link's most common misuse has a name, fixed by this method and unchanged here, and mapped to the part of the chain that guards against it.

d-VISION — buzzword vision. A vision that names all five domains of change and scores none of them honestly: every clause reads well and none converts into a falsifiable observation. Guarded at L2 — a vision this challenge set would fail is not a shorter vision; it is the same words with the decision still missing underneath them.

d-TECH — technology first. An enabler decision made before the operating model has named what it should power, so the platform is chosen and the case for it is written afterwards. Guarded across L4–L7, where every enabler cites the operating-model dimension it serves; a technology decision with no cited dimension upstream of it has jumped the queue, whatever budget line already exists for it.

d-OCEAN — boiling the ocean. Every initiative attempted in the same wave, so the initiatives that unblock the rest receive the same starved attention as the ones still blocked behind them, and none reaches readiness. Guarded at L8, where staging, not ambition, is the discipline: a roadmap with one wave is not a bolder roadmap, it is an unsequenced list with a date attached.

d-PILOT — pilot purgatory. A pilot that proved an idea inside conditions the rollout will not have, then rolled out unchanged. Scaling was treated as copying, rather than as the operating-model change the idea still needs. Guarded at L9, where the metric tree ties a leading number to the people and process dimensions a pilot's own bubble never tested. A metric that only ever measured the pilot's own conditions cannot show the rollout will hold.

d-TOM — no operating-model change. A strategy can show a data workstream, a technology workstream and an AI workstream, and no people, process or governance workstream. Three columns look complete, yet on the one dimension the whole chapter exists to force, that is digitalisation wearing a transformation's name. Guarded at `L3` and again at `L10` — the chapter's own climactic failure. Every anti-pattern above can survive one bad Link; this one survives the whole chain if the bridge and the governance synthesis are not both checked.

7.5 What this paper does not claim

Strategy is not a plan. A strategy is a set of choices; a plan is the scheduling, resourcing and sequencing of action against a strategy already formed. Chapter 3 makes the choices — which operating model, which enablers, in what sequence, what counts as success — bounded throughout by the pillars it inherits. It hands the result, unscheduled and unresourced, on to Chapter 4.

Mintzberg's own verified language supports the distinction without needing to overstate it: "strategic planning is not strategic thinking" (Mintzberg, 1994, *Harvard Business Review*, "The Fall and Rise of Strategic Planning," pp.107–114). Conventional planning, on his account, does well what he calls strategic programming — "the articulation and elaboration of strategies, or visions, that already exist" — never the forming of the strategy itself. That programming role belongs to Chapter 4: mobilisation, scheduling, budgeting, running GC3. Chapter 3 sits upstream of that seam, with Chapter 2, where the choices are made, and this passage states that boundary once.

Mintzberg and Waters (1985) place real strategies, in their own words, at "two ends of a continuum along which real-world strategies lie," not at either pole alone. A strategy bounded by the pillars it inherits behaves less like a rigid schedule and more like what they term an umbrella strategy — central guidelines within which detail is free to emerge. That is an analogy this chapter borrows, not a finding it claims about itself. Grant (2003), studying eight oil majors, found that formal planning systems did not disappear under turbulence; their job changed to what he calls "planned emergence" — coordinating decentralised strategy-making inside guardrails and demanding performance targets. Chapter 3's staged roadmap, and the revision conditions it names for each wave, work the same seam. Chapter 3 defines the guardrails a wave must clear; the governance that runs them sits downstream, with Chapter 4.

A written, coded, falsifiable strategy is a formal artefact. Nothing in Mintzberg's argument says that artefact cannot record a choice someone's judgement already made — only that it must not be mistaken for the act of making it.

No claim of better decisions or outcomes. No study shows that a strategy produced by this chain leads to better decisions or a higher transformation success rate. None compares it against a strategy produced by an unstructured or differently structured approach. The chain's claim is limited to the auditability and traceability of the strategy it produces, never to what the strategy goes on to achieve.

No claim of originality for the underlying frameworks. The maturity model, the five domains of change and the capability-based bridge to an operating model are each cited to established literature. So are the data-offence/defence balance, the product operating model and the decision-rights archetypes. The decision-rights archetypes are drawn from Weill and Ross (2004). The chain's own contribution is operationalising and sequencing them into one coded chain, not inventing any one of them.

No numeric AI-readiness threshold. No standards-grade source states one; the closest published analogue evaluates public-sector AI deployment, not enterprise transformation, and is itself an unreviewed preprint. The gap is stated as a gap, not filled with an invented number. The maturity read at the chain's own first Link carries a matching limitation: no numeric admission threshold either, a documented category of maturity-model weakness in the wider literature (Andersen et al., 2020; Becker et al., 2009), not a shortcut unique to this chain.

No research-grade scoring rubric. Placement on the value-and-feasibility grid and selection of a North Star metric are both undocumented-origin practitioner conventions. No primary source specifies how either should be scored — which is exactly why challenges 4 and 5 above exist, to test a placement or a measure by argument, since no formula settles it.

No statistic behind the operating-model argument, and no failure-rate hook. The case for changing all five operating-model dimensions at once rests on the chain's own logic, never on a cited statistic. A model changed on one dimension is not the model the vision requires. Where a failure-rate figure appears anywhere in the wider paper, it is limited to a single admissible finding on sustained performance improvement, never a splice of unrelated studies into one round number.

No benchmark on outcomes. The comparison the wider paper draws against a capability-architecture method, an operating-model archetype study, an open five-domain lens or a staged consultancy pattern is one of coverage — what each names and sequences — never of results (the no-benchmark non-claim above states the full case once).

No claim that Chapter 3 performs Chapter 4's programming. "Roadmap" names L8's own staged artefact, never the chain's output as a whole, which is handed on unscheduled to the chapter that mobilises, schedules, budgets and runs it.

8. Applicability

Ten Links, an entry contract and an exit contract cost more than a five-dimension slide and a staged roadmap chart. That cost buys something specific — a pillar that only needs a technology refresh does not owe it.

Use it when

- The pillar or pillars admitted from Chapter 2 imply a change to more than one of process, technology and the organisation underneath it. The chain's own climactic failure mode is a technology change with nothing else moving, and only a forced five-dimension check catches that quietly rather than after a wave is funded.

- More than one enabler is implicated — data, AI, technology, governance design. A single-dimension enabler decision, taken without the sequencing law, risks starting an AI use case ahead of the data foundation it depends on.
- The organisation is willing to name, before it commits a wave in detail, the metric and the threshold that must clear before the next wave starts, and the assumption that threshold tests. The chain's own answer to reality disagreeing with the strategy is a named revision condition, not a silent rewrite.
- A pillar's assumption could plausibly be shown wrong once work is under way. The organisation wants a standing, formal route to say so, rather than quietly patching the vision or the operating model to fit new evidence.
- Decision rights, a funding model and a review cadence need to be designed and agreed before Chapter 4 is asked to run them. Who decides should be settled before the first quarterly re-sequencing meeting, not during it.

Lighten it or choose another approach when

- The change is confined to one dimension — a technology swap with no process, structure or governance consequence. Running the full five-dimension check and the enabler grid then costs more than the change is worth.
- No enabler touches AI, so the data-before-AI sequencing law, and the gating dependency it forces on the AI cell, add nothing the organisation does not already know to do.
- The organisation cannot or will not commit to a numeric baseline, target and date for its metrics, or to a threshold value for a wave transition. Running the roadmap and metrics Links without either produces a chain that looks complete and states nothing falsifiable.
- The pillar or pillars admitted from Chapter 2 are thin, scoped to a single team with no maturity read worth taking, and the entry contract's own clauses are the more defensible place to stop.

Preconditions

Before the entry contract runs, confirm:

1. at least one Chapter 2 pillar able to meet the entry contract's four clauses, or a clear route to get one there;
2. standing to name decision rights and a funding model spanning more than one wave, not a single budget line;
3. willingness to state, before the roadmap Link runs, the metric and threshold values the organisation's own instance of the method will use — this chapter fixes none of its own beyond a generic illustration;
4. a route back to Chapter 2's own stop 15 that is genuinely available, not merely notional, for a pillar that fails admission or is later falsified.

An organisation that wants a shaped, sequenced, governed strategy by the end of the quarter and a chain that clears every precondition are not always compatible. The transparent response is to say so before starting, and to scope down deliberately — fewer Links run in full, more pillars referred early. That is better than clearing a precondition quietly and calling the strategy complete.

9. Why this chain and not the alternatives

Four established bodies of work address pieces of the same problem — moving from a strategic choice to a governed digital transformation — and none is rejected on preference. Each is compared against what the ten-Link chain adds, and where a comparator was not itself independently re-opened for this chapter, that limitation is stated rather than smoothed over.

Against TOGAF's Architecture Development Method. The Open Group's own standard text sits behind a login wall this research could not clear, so nothing here rests on its primary wording. The shape described is corroborated only through a secondary account, cross-checked against its own citations. At that level, the ADM is an iterative method for developing and managing an enterprise-architecture life cycle, with business-capability planning named as a phase before technology design. That is a genuine precedent for "capability gaps before an operating model," independently echoed by the capability-based framing below. What it does not publish, at any level this research could verify, is a digital-transformation-specific diagnostic, a forced falsifiable-vision derivation, or a coded, per-artefact chain from a strategic choice through to a governed, metricated handoff.

Against MIT CISR's Future Ready pathways. As presented in Woerner, Weill and Sebastian (2022), research drawn from a panel of more than a thousand companies selects among operating-model archetypes and names a target vocabulary of future-ready capabilities. It operates at the level of which pathway an organisation should choose, not at the level of which initiative to sequence into which wave. As publicly described, the framework does not set out a governance-cadence design contracted to a named downstream handoff.

Against Rogers' Digital Transformation Playbook. The five domains this chain's own vision Link borrows — customers, competition, data, innovation, value — are offered as an open analytic lens for finding where digital changes the game. They are not offered as a forced sequence with a falsifiable-sentence exit test. A team may use the five domains as a checklist and stop; the chain's own vision Link cannot stop until its coded sentence traces to named domains and states what would make it false.

Against major-consultancy staging patterns. More than one large consultancy publishes a foundation-then-scale-then-differentiate staging pattern for a transformation portfolio, a shape this chain's own three waves echo. Those reports were not independently re-opened for this pass, so no specific claim from either is repeated here. The comparison is limited to the shape of the staging, not to any stated outcome.

What the chain adds, independently, against each comparator's gap.

1. A fully coded, per-artefact chain from the admitted pillar set through to a governed, metricated strategy, with every artefact's upstream parents explicit — none of the four comparators publish a chain at this grain.

2. A named design/run governance boundary, contracted to a specific downstream handoff rather than left as a general concept. An Evaluate-Direct-Monitor split supports the idea that design and run are distinct, but none of the four comparators contract that split to a named artefact the way this chain's final Link does.
3. A forced, falsifiability-tested vision sentence, with a stated pass condition, rather than an open lens a team may or may not apply to a conclusion.

What is not claimed. No study shows this chain, or any ordering of a digital-transformation method, produces better decisions or a higher transformation success rate than a comparator above (§7.5 gives the paper's one full statement of this non-claim). The case for ten Links over a shorter or differently-shaped sequence rests on what each admission check and each contract check is shown above to add, not on a measured result.

Extended rationale: foundations and the entry-contract contribution

Capgemini Consulting and the MIT Center for Digital Business's (2012) study places digital intensity against transformation-management intensity across four named quadrants; its further findings on which quadrant outperforms another are not this chapter's evidence, and no numeric threshold on either axis exists in the source, so L1 borrows the two-axis shape only, never a scored cut-off. Rogers (2016) offers the five domains as a practitioner lens built from case research, not a controlled study of whether mapping to them outperforms another framing; this chapter cites it on those terms only, and reads an untouched domain as a named gap, never a silent drop. Weill and Woerner (2018) map a different archetype fork — Supplier, Omnichannel, Modular Producer, Ecosystem Driver — from which this chapter borrows only the observation that a firm's orientation is a two-dimensional choice; the specific fork L2 forces is this method's own convention, never a finding of that book or of MIT CISR's.

The entry contract's own contribution is likewise bounded. An admission check, a narrow verdict vocabulary and a self-naming referral are each established practice in some form; the contribution is holding that practice to the two-sided discipline Chapter 1 established and Chapter 2 runs, at a seam neither chapter's own contract reaches. No statistic underwrites the entry contract, and no benchmark measures it against an alternative: it offers only that an admitted pillar carries a coded parent, a resolvable source, a stated consequence and a named scope before a single Link ran against it, never that carrying those four made the pillar itself the right one.

10. Scope, roadmap and references

10.1 Scope of Chapter 3

Chapter 3 specifies the digital-transformation chain: the entry contract on Chapter 2's pillars, the four acts and ten Links from digital maturity through to governance detail and synthesis, and the HX1→HX2 exit contract. Its register marks every rule MUST, SHOULD or MAY; the method fixes no count of rules. It is written as a reference against which a digital-transformation strategy can be produced and then audited — what each Link admits, what it refuses, what it produces. It also names the condition its output must meet before the next Link, or Chapter 4, may rely on it.

The worksheets, scoring and a full worked case that put the chain into practice sit in the practitioner guide, not here.

It does not cover:

- producing or repairing a pillar's content, which stays Chapter 2's, at either the entry-time or the post-admission referral (§3.2);
- what follows the handoff: detailed delivery planning, the risk register, change-management and adoption mechanics, running the governance cadence, and in-flight benefits tracking — all Chapter 4's;
- evidence that ten Links outperform a shorter, longer or differently-shaped sequence: the chain is reasoned, not empirically benchmarked (§9).

Illustrations throughout are generic and informative: no named company, no fixed count beyond the method's own fixed sets — four acts, ten Links, two entry-contract verdicts, eight challenges.

10.2 Roadmap

The execution chain — the fourth chapter, carrying the work from mobilisation to benefits realisation, where this chain's HX1 pack hands off at the HX2 boundary — is published as Chapter 4, The Execution Chain. The following remain planned, with timing set by the author's prioritisation, each keeping the rule "No output without a traceable input".

- **Practitioner guide** — worksheets, a scoring rubric and a full worked case for the ten Links and the eight challenges, in preparation.
- **A later revision of this chapter** — the eight challenges are the ceiling as this chapter stands. A ninth would need its own evidence, not a preference, and would be recorded as a new changelog entry, not folded into this one.

This fills the slot Chapter 2's own roadmap already reserves for the digital-transformation chain. Diagnosis and the analysis chain that precede it are Chapter 1 and Chapter 2, already published.

10.3 Changelog

Date	Change
22 September 2026	**First published.** The entry contract, four acts and ten Links — digital maturity, digital vision, target operating model, the four-part enabler grid, roadmap, metrics, and governance detail and synthesis — the rule register, the eight challenges, and the HX1→HX2 exit contract.
28 September 2026	**Restructured for reading order, no rule changed.** The exit contract (§6.5) moved to §5.11, immediately after L10, and the eight challenges (§7.3) moved to §5.12, immediately after it; §1.2's RFC 2119/8174 boilerplate replaced with a pointer to the shared conventions page (whitepaper.strategyascode.org/conventions/); §2 Foundations and §3.4 were condensed to a front residual, with the fuller argument moved into §9's new "Extended rationale" sub-heading; the no-benchmark non-claim, previously stated four times, now has one full statement (§7.5), with the Abstract, §9 and §7.5's own second occurrence pointing to it; the orientation-fork duplicate bullet was removed from §7.5 (the same claim survives at §5.12's Challenge 8 and at §2); §1.1, §1.3, §3.1, §3.2 (non-rule prose only), §3.3, §4.1, §4.2, §4.3, §6.3, the Abstract and About this document were condensed or tightened. No Link, admission check, contract check, challenge or verdict was added, removed or reworded, and every rule sentence keeps its original wording; the exit contract's figure caption and SVG <title> were renumbered Figure 5.9 (previously 6.3) to match its new location.
28 September 2026	**§7.1 wording corrected to point to the challenge set's new location (§5.12); no rule changed.**
28 September 2026	**Version numbers removed; changes are now recorded by date; no rule changed.**

This changelog belongs to Chapter 3 alone; it leaves Chapter 1 and Chapter 2's own changelogs untouched. A future change to this chapter is recorded as a new entry here, never as a silent edit to this one.

10.4 References

- Andersen, Lee, Mettler & Moon, "Ten Misunderstandings about Maturity Models" (Proceedings of the 21st International Conference on Digital Government Research, 2020, pp. 261–266) — https://doi.org/10.1145/3396956.3396980
- Becker, Knackstedt & Pöppelbuß, "Developing Maturity Models for IT Management" (*Business & Information Systems Engineering* 1(3), 2009, pp. 213–222) — https://doi.org/10.1007/s12599-009-0044-5
- S. Bradner, "Key Words for Use in RFCs to Indicate Requirement Levels" (BCP 14, RFC 2119, RFC Editor, 1997) — https://www.rfc-editor.org/rfc/rfc2119
- M. Cagan / Silicon Valley Product Group, "The Product Operating Model: An Introduction" — https://www.svpg.com/the-product-operating-model-an-introduction/
- Capgemini Consulting and MIT Center for Digital Business, *The Digital Advantage: How digital leaders outperform their peers in every industry* (2012) — https://ide.mit.edu/sites/default/files/The_Digital_Advantage__How_Digital_Leaders_Outperform_their_Peers_in_Every_Industry.pdf
- L. DalleMule and T. H. Davenport, "What's Your Data Strategy?" (*Harvard Business Review*, May–June 2017) — https://hbr.org/2017/05/whats-your-data-strategy
- R. M. Grant, "Strategic Planning in a Turbulent Environment" (*Strategic Management Journal* 24(6), 2003, pp. 491–517) — https://doi.org/10.1002/smj.314
- B. Henderson, "The Product Portfolio" (BCG *Perspectives*, 1970) — https://www.bcg.com/publications/1970/strategy-the-product-portfolio
- ISACA, *COBIT 2019 Framework: Governance and Management Objectives* — https://www.isaca.org/resources/cobit
- Legara, Jose & Martinez, "Beyond Model Readiness: Institutional Readiness for AI Deployment in Public Systems" (arXiv:2605.17203, 2026) — https://arxiv.org/abs/2605.17203
- B. Leiba, "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words" (BCP 14, RFC 8174, RFC Editor, 2017) — https://www.rfc-editor.org/rfc/rfc8174
- E. A. Locke and G. P. Latham, "Building a Practically Useful Theory of Goal Setting and Task Motivation: A 35-Year Odyssey" (*American Psychologist* 57(9), 2002, pp. 705–717) — https://doi.org/10.1037/0003-066X.57.9.705
- MACH Alliance, official site — https://machalliance.org/
- H. Mintzberg, "The Fall and Rise of Strategic Planning" (*Harvard Business Review* 72(1), 1994, pp. 107–114) — https://hbr.org/1994/01/the-fall-and-rise-of-strategic-planning
- H. Mintzberg and J. A. Waters, "Of Strategies, Deliberate and Emergent" (*Strategic Management Journal* 6(3), 1985, pp. 257–272) — https://doi.org/10.1002/smj.4250060306
- D. L. Rogers, *The Digital Transformation Playbook* (Columbia Business School Publishing, 2016) — https://cup.columbia.edu/book/the-digital-transformation-playbook/9780231175449/
- "TOGAF" (Wikipedia, cross-checked secondary source; the standard's own text was not independently opened this pass) — https://en.wikipedia.org/wiki/TOGAF
- P. Weill and J. W. Ross, *IT Governance* (Harvard Business School Press, 2004) — https://archive.org/details/itgovernancehowt0000weil
- P. Weill and S. L. Woerner, *What's Your Digital Business Model?* (Harvard Business Review Press, 2018) — https://cisr.mit.edu/publication/whats-your-digital-business-model

- S. L. Woerner, P. Weill and I. M. Sebastian, *Future Ready: The Four Pathways to Capturing Digital Value* (Harvard Business Review Press, 2022) — https://cisr.mit.edu/publication/future-ready-four-pathways-capturing-digital-value

Appendix A — The Links and their rules

Each Link's admitted input, produced output and contract check are given once, in full, at §4.3; this appendix does not repeat them. One row per Link instead states the rules that carry the chain's normative weight, each marked by its own level, stated by content and never by an internal identifier. Every Link runs inside the entry contract on Chapter 2's pillars and the HX1→HX2 exit contract that names the consolidated pack a digital-transformation strategy.

Link	Key rules
L1 Digital Maturity	The maturity placement is diagnostic colour, never a MUST-level admission threshold; no numeric threshold for it exists in the literature. The verdict MUST name its two axis readings and their pillar sources.
L2 Digital Vision	DVX MUST derive only from the orientation and three of the five domain reads, never the other two; the three forming steps SHOULD be forced, never brainstormed; DVX MUST be falsifiable and every clause MUST trace to a named domain code; all five domains SHOULD be named even when one reads amber; the orientation is this method's own convention, not an external finding.
L3 Target Operating Model	The operating model MUST change on all five dimensions at once; a technology-only change is digitalisation, not transformation — the chain's strongest rule. Every capability MUST cite a domain or pillar parent, and all five dimensions MUST be populated or the gap explicitly named.
L4 Data	Data is the enabler the sequencing law starts from; each cell MUST record one decision, never a shopping list — shared by all four cells, with no cell-specific rule beyond it.
L5 AI	AI is the gated cell: EN–A MUST name its gating dependency on the data enabler explicitly, and MUST NOT start before the data foundation exists; no quantified threshold for "exists" is claimed.
L6 Technology	Composability is an industry-consortium convention, not an academically sourced rule; the cell MUST record one decision, never a shopping list, with no cell-specific rule beyond it.
L7 Governance-design	Scoped to design only, never to running the model; feeds L10's governance detail directly; the cell MUST record one decision, never a shopping list, with no cell-specific rule beyond it.
L8 Roadmap	Every I# MUST cite a parent — an enabler decision or a depended-on initiative — reaching an enabler unbroken; a candidate with no parent MUST be refused outright. Placement SHOULD be defended against one named alternative, never MUST-level; enablers MUST reach readiness before use-cases scale, and an AI initiative MUST NOT precede the data foundation it depends on; an initiative MUST NOT be staged in the same wave as, or earlier than, any initiative it depends on — the grid ranks, dependencies stage; each wave transition MUST name its clearing metric, threshold value and tested assumption — Chapter 3 defines these conditions, Chapter 4's GC3 stage-gate runs them.
L9 Metrics	Every K target SHOULD be numeric with a named baseline, target and date; K0 MAY serve as the North-Star mechanism, which carries no academic or standards-body origin.
L10 Governance detail & synthesis	GOV3 SHOULD state its cadence, inputs, decision rights and escalation path in full; the funding model releases each wave's budget only against K0 movement, with K3 tracked as the target it watches; Chapter 3 designs the cadence, never runs it.

As §6 states, every path through the chain's own graph reaches the HX1 pack; the graph itself closes one step further, at the boundary marker HX2, which cites HX1 alone. HX2 names what Chapter 4 owns and what it does not, so that the strategy handed off is bounded as precisely as it is coded.

Appendix B — The derivation register

This appendix registers the illustrative instance Figure 6.1 schematises: all fifty-two codes, grouped by the Link that mints each one, together with the parents each code cites. Five edges are marked inferred (§6.3); a pillar's row carries no parent, since it enters the chain rather than deriving from it.

Link	Code	Cites (parents)
Entry contract	P·1	— (entry)
Entry contract	P·2	— (entry)
Entry contract	P·3	— (entry)
Entry contract	P·4	— (entry)
L1 Digital Maturity	DM1	P·1, P·2
L1 Digital Maturity	DM2	P·3
L1 Digital Maturity	DM3	DM1, DM2

Link	Code	Cites (parents)
L1 Digital Maturity	DM4	—
L2 Digital Vision	OR–E	P·1, P·2
L2 Digital Vision	OR–S	P·4
L2 Digital Vision	DV1	OR–E, P·1
L2 Digital Vision	DV2	P·2
L2 Digital Vision	DV3	P·1, P·3
L2 Digital Vision	DV4	P·4
L2 Digital Vision	DV5	DM1
L2 Digital Vision	DVX	DV1, DV2, DV3, OR–E
L3 Target Operating Model	CAP·1	DV1, P·1
L3 Target Operating Model	CAP·2	DV2, P·2
L3 Target Operating Model	CAP·3	DV4, P·1
L3 Target Operating Model	CAP·4	P·3
L3 Target Operating Model	CAP·5	DV5
L3 Target Operating Model	OM–P	CAP·4, P·3
L3 Target Operating Model	OM–R	CAP·1, CAP·2
L3 Target Operating Model	OM–D	CAP·2, DV2
L3 Target Operating Model	OM–T	CAP·3, DV4
L3 Target Operating Model	OM–G	CAP·4
L4 Data	EN–D	DV2, OM–D
L5 AI	EN–A	CAP·2, EN–D
L6 Technology	EN–T	CAP·3, OM–T
L7 Governance-design	EN–G	CAP·4, OM–G
L8 Roadmap	I1	EN–D (inferred)
L8 Roadmap	I2	EN–T (inferred)
L8 Roadmap	I3	EN–G (inferred)
L8 Roadmap	I4	I1, I2, I3
L8 Roadmap	I5	I1
L8 Roadmap	I6	EN–A (inferred), I1, I2
L8 Roadmap	I7	EN–A (inferred), I1, I5
L8 Roadmap	I8	I4
L8 Roadmap	RW1	EN–D, EN–G, EN–T, I1, I2, I3
L8 Roadmap	RW2	EN–A, I4, I5, RW1
L8 Roadmap	RW3	CAP·5, EN–A, I6, I7, I8, RW2
L9 Metrics	K0	DVX, P·1, K1, K2
L9 Metrics	K1	OM–D, P·2
L9 Metrics	K2	OM–P, P·3
L9 Metrics	K3	K0, K1
L9 Metrics	K4	P·4, RW3
L9 Metrics	K5	EN–G, OM–R
L10 Governance detail & synthesis	GOV1	EN–G, OM–G
L10 Governance detail & synthesis	GOV2	K0, RW1

Link	Code	Cites (parents)
L10 Governance detail & synthesis	GOV3	GOV2, RW1
L10 Governance detail & synthesis	HX1	DVX, GOV1, GOV2, GOV3, K0, K1, K2, K3, K4, K5, OM-D, OM-G, OM-P, OM-R, OM-T, RW1, RW2, RW3
L10 Governance detail & synthesis	HX2	HX1

About the author

Aliaksei "Alex" Ivanouski is a Fractional Chief Technology Officer (CTO) and founding engineer with 10+ years building fintech, adtech and backend products. A military-engineer-turned-founder, he holds a Master of Business Administration (MBA) with Distinction from Bayes Business School, City St George's, University of London, and completed the CTO Programme at the National University of Singapore. He built the Business Strategy as Code method and applies it in a practitioner programme at strategyascode.org.

Colophon

- **Status.** Published 22 September 2026; last updated 28 September 2026.
- **Production.** Generated from the author's source text.
- **How to cite.** Ivanouski, A. (2026). *Business Strategy as Code: A Governed Method for Digital-Transformation Strategy*, Chapter 3. https://whitepaper.strategyascode.org/digital-transformation-chain/. Chapters 1, 2 and 4 are cited the same way, each with its own chapter number, title and URL.
- **Updates.** Changes to a chapter are recorded, dated, in its own §10.3; earlier wording is not silently altered, and one chapter's changelog never affects another's.