

ENTERPRISE AGILE ENABLEMENT · V2.0

Practitioner Playbook

study bible + leadership implementation for agile coaches stepping into
enterprise delivery enablement

Print-ready field guide for regulated and federated enterprises

For coaches, transformation leads, delivery directors, and Agile CoE practitioners

v2.0 · June 2026

Confidential working document · Written in British English · Companion to the Executive Brief

PART I — STUDY BIBLE

1. Mindset shift: team coach to enterprise lead	5
2. Three layers: operating model, enablement system, rollout plan	7
3. Capability map with scoring table	9
4. 30/60/90 learning plan	12
5. Four operating model sketches A-D with trade-offs	15
6. Six needs scoring framework	18
7. Framework mix: Scrum, SAFe, Flight Levels	20
8. Red flags vs strong language	22
9. Stakeholder map: what each group needs to hear	24

PART II — CASE EVIDENCE

10. Case A: tier-1 European bank, multi-country liquidity reporting	28
11. Case B: greenfield enterprise Agile CoE	31
12. Failure modes: signals and responses	34

PART III — LEADERSHIP PLAYBOOK

13. Eight-step rollout: timeline and owners	37
14. Step 1 Diagnose: checklist and outputs	40
15. Step 2 Operating model: sketch questions and workshop prep	43
16. Steps 3-8 summaries: practical action lists	46
17. Leadership scripts with objection handling	49
18. SteerCo narrative: the four questions leaders ask	53
19. Metrics dictionary: team, programme, enterprise	56
20. Business case framing	60
21. First 90 days week by week	63
22. Year 1 success criteria	66

PART IV — ROLE READINESS

23. Interview preparation	69
24. What to do when the pilot fails	73
25. Anti-pattern watchlist	76

HOW TO USE THIS DOCUMENT

Read Part I before you take the role, Part II when you need proof that the approach survives enterprise reality, Part III when you are in front of sponsors, and Part IV when you are preparing for selection, transition, or recovery.

This section is designed for practitioners who already know how to coach teams and now need to operate credibly at enterprise level. The shift is not about learning a larger ceremony catalogue. It is about learning to design a federated delivery system, tell the leadership story for it, and make difficult trade-offs explicit before a rollout turns into theatre.

At team level you improve a local system. At enterprise level you design the conditions that allow dozens of local systems to deliver one shared outcome without collapsing into politics, rework, or audit surprise.

1. Mindset shift: team coach to enterprise lead

Team coaching rewards immediacy. You can observe a sprint, see a weak retrospective, improve backlog refinement, and watch the next iteration move. Enterprise enablement moves at a different altitude. The problem is rarely that people do not know a ceremony. The problem is that the organisation has never made explicit decisions on decision rights, planning horizons, cross-entity hand-offs, regulatory capacity, and what is mandatory versus local.

An enterprise lead must therefore be bilingual. You still need enough team-level fluency to know whether the proposed operating model can work in practice, yet your real product is no longer a healthy team. Your real product is a delivery system that leaders can understand, governance partners can trust, and teams can operate without constant exceptions.

Dimension	Team coach	Enterprise enablement lead
Primary unit of change	One squad, one product team, or a small cluster of teams	Value stream, programme, business unit federation, or enterprise capability system
Definition of success	Healthy ceremonies, clearer backlog, stronger ownership, better collaboration	Adoption of a target operating model, improved flow across boundaries, visible governance, better predictability
Typical timescale	Sprint to sprint, or over a quarter	Quarter to year, through pilot and rollout waves
Core conversations	Scrum Master, Product Owner, engineers, delivery manager	Sponsor, business unit lead, enterprise architect, compliance, HR, portfolio office, internal coaches
Failure mode	Local improvements fade when the coach leaves	Transformation spends heavily but never institutionalises shared ways of working
Language that lands	Team health, collaboration, prioritisation, impediment removal	Operating model, decision rights, dependency flow, adoption, regulatory exposure, leadership choices

MINDSET YOU LEAVE BEHIND

"If teams improve, the system improves."

This is sometimes true in small organisations. In a federation, excellent teams can still be trapped by funding gates, architecture queues, shared service bottlenecks, and incompatible planning rhythms.

MINDSET YOU ADOPT

"If the system is explicit, teams can improve inside it."

Your work is to define the scaling backbone, governance hooks, and enablement paths that let local teams work productively without each business unit improvising its own approach.

PRACTICAL IMPLICATION

When leadership asks, "How do we scale Agile?", do not answer with a framework recommendation. Answer with the next enterprise decision: what operating model choices must be made before any framework is selected or training is purchased.

2. Three layers: operating model, enablement system, rollout plan

Enterprise Agile succeeds when three layers work together. Missing any one of them creates a recognisable failure pattern. Organisations without an operating model devolve into framework arguments. Organisations without an enablement system rely on charismatic individuals and expensive consultants. Organisations without a rollout plan create an impressive pilot that never survives contact with the rest of the estate.

Think of the three layers as design, capability, and expansion. The operating model defines how work should flow. The enablement system creates the people, forums, and routines that make the model operable. The rollout plan sequences adoption so that early wins become institutional assets rather than isolated anecdotes.

Layer 1 · Operating Model

Decision rights · scaling units · planning rhythm · governance hooks

Layer 2 · Enablement System

CoE mandate · role paths · coach network · leadership forums · measurement

Layer 3 · Rollout Plan

Pilot slice · wave sequencing · adoption criteria · transition to internal ownership

LAYER 1

Operating model

Defines who decides what, how work is grouped, which planning cadence binds the enterprise together, and where governance enters the flow.

LAYER 2

Enablement system

Creates the reusable machinery: CoE charter, role expectations, internal coaching, leadership forums, and written standards that survive personnel change.

LAYER 3

Rollout plan

Determines where to start, how to sequence adjacent areas, what metrics prove readiness, and how external support hands over to internal ownership.

Layer	Key question	Typical artefacts	What goes wrong when missing
Operating model	How should enterprise delivery work here?	Model sketch, decision record, governance map, mandatory vs local standards	Each area defines Agile differently; leaders cannot compare or steer
Enablement system	Who enables, teaches, audits, and improves the model?	CoE charter, RTE/PO/leader pathways, internal coach plan, cadence calendar	Knowledge remains with consultants; practices decay after the pilot
Rollout plan	How do we move from one pilot to enterprise adoption?	Pilot charter, wave roadmap, entry and exit criteria, adoption dashboard	One excellent reference case, no repeatability, constant reinvention

The mistake is not starting with SAE. The mistake is starting with anything before you know the operating model, the enablement system that sustains it, and the rollout path that makes it repeatable.

3. Capability map with scoring table

Before taking the role, score yourself honestly. Enterprise delivery enablement is unforgiving to practitioners whose strengths are purely facilitative. You do not need to be a deep expert in every adjacent function, but you do need working fluency across the whole system. The scoring table below is intended as a diagnostic, not a confidence exercise.

Use a five-point scale: **1** means you understand the terms but could not yet lead the work; **3** means you can perform independently with preparation; **5** means you can teach, adapt, and recover the topic under executive pressure.

Capability	1	2	3	4	5	What good looks like
Federated system thinking						Can map business units, countries, shared services, and decision rights on one page without losing the plot.
Operating model design						Can sketch multiple viable target models, articulate trade-offs, and recommend one with a fallback.
Executive narrative						Can explain enterprise Agile in terms of outcomes, risk, governance, and investment rather than ceremonies.
Programme design						Can structure ARTs or equivalent groupings, define PI rhythms, and recognise when boundaries are wrong.
Flow visibility						Can use Flight

						Levels or similar concepts to show leadership where work stalls above team level.
CoE design and mandate						Can define authority, services, staffing model, rhythms, and transition plan for an Agile CoE.
Regulated delivery						Can embed compliance, architecture, and audit evidence in the normal delivery flow rather than a side channel.
Change sequencing						Can decide where to pilot, how to wave rollout, and when a unit is genuinely ready to scale.
Team coaching baseline						Still able to judge whether team-level practice is credible enough to sit on the enterprise backbone.

INTERPRETATION RULE

Any score below 3 in federated system thinking, operating model design, executive narrative, or regulated delivery should be treated as a priority development gap before you position yourself as an enterprise lead.

How to improve your weak spots

IF DESIGN IS WEAK

Draw models, do not only read about them

Take one real enterprise case, sketch three operating models, and force yourself to choose one with explicit trade-offs.

IF NARRATIVE IS WEAK

Translate Agile into board language

Practise explaining PI predictability, cost of delay, and regulatory capacity without using ceremony jargon.

IF REGULATION IS WEAK

Study governance in flow

Map audit evidence, sign-off, and risk review into the work system so that compliance is neither bypassed nor delayed until the end.

4. 30/60/90 learning plan with study, practice, and evidence

Practitioners often ask what to learn first. The answer is not "more frameworks". The answer is a staged learning plan that develops system thinking, enterprise language, and implementation discipline in that order. The table below assumes you are preparing for an interview or a new role and can devote structured effort each week.

Window	Study	Practice	Evidence you can show
Days 1-30	Read one enterprise case in detail. Learn the anatomy of value streams, ARTs, federated governance, and portfolio constraints. Study why operating models fail when leaders leave choice ambiguous.	Map one target organisation. Draft a current-state federation map and identify three systemic bottlenecks above team level.	One-page federation map, list of bottlenecks, and a five-minute explanation of why the organisation should not begin with training.
Days 31-60	Study scaled programme design, role architecture, and flow visualisation. Learn enough Flight Levels to speak credibly about work above team boundaries.	Create four operating model sketches for the same enterprise context. Score them against six enterprise needs. Facilitate a mock selection discussion.	Sketch pack, scoring matrix, and a draft decision record with target model plus fallback.
Days 61-90	Study rollout sequencing, internal capability building, and transformation anti-patterns. Learn what good adoption metrics look like in regulated organisations.	Draft a pilot charter, a CoE charter, and a ninety-day leadership plan. Rehearse sponsor, BU, and compliance conversations.	Pilot charter, CoE charter, first-90-days plan, and a concise executive narrative you can present under challenge.

Recommended weekly rhythm

Cadence	Activity	Purpose
Twice weekly	Thirty-minute concept study block	Deepen vocabulary in operating models, governance, and flow
Weekly	Ninety-minute sketching session	Turn theory into concrete models, artefacts, and talk tracks
Weekly	Executive narrative rehearsal	Practise answering "why this, why now, what decision do you need from me?"
Fortnightly	Peer challenge review	Expose weak assumptions and jargon before a real leadership setting does

EVIDENCE BEATS ASPIRATION

By day 90 you should have tangible artefacts: model sketches, a scoring framework, a pilot plan, and rehearsed scripts. Saying "I understand enterprise Agile" is weak. Showing the artefacts you would use in week one is convincing.

5. Four operating model sketches A-D with trade-offs

No single operating model fits every enterprise. The purpose of the four sketches is not to find a universal answer, but to train yourself to recognise the dominant constraint in front of you. In practice, most enterprises use one primary model plus one overlay. The most common mistake is pretending that local autonomy, regulatory certainty, and standardisation can all be maximised simultaneously.

SKETCH A

Central CoE with federated ART backbone

Best where the organisation needs a common cadence, common metrics, and cross-country consistency. Useful in banking, insurance, and energy groups with strong central risk or reporting obligations.

SKETCH B

BU-led clusters with light central standards

Best where business units have meaningful autonomy and political resistance to central design. It buys faster adoption but weakens comparability and cross-enterprise steering.

SKETCH C

Shared enablement spine around platform bottlenecks

Best where architecture, data, platform, or security services are the real bottleneck. The model emphasises visible cross-functional flow above team boundaries.

SKETCH D

Regulatory overlay on top of A or C

Best where a hard deadline or reporting obligation dominates capacity. It should not replace the main model; it should shape how capacity, sign-off, and evidence are managed.

Sketch	Good fit when	Strengths	Trade-offs	Common mistake
A	Enterprise needs one visible backbone across countries or BUs	Consistency, clearer governance, easier group reporting, stronger internal role standards	Slower local tailoring; requires sponsor backing and disciplined exceptions handling	Announcing standardisation without funding the CoE and role enablement required to sustain it
B	Local P&L power is high and forcing a central model would trigger rejection	Political fit, faster local uptake, room for experimentation	Weak enterprise transparency, inconsistent roles, duplicated learning, difficult audit story	Calling it federated by design when it is actually fragmented by default
C	Shared services delay everything and product teams cannot move without them	Targets the real bottleneck, strengthens dependency management, improves architecture dialogue	Needs mature facilitation and flow visibility; can feel abstract to leaders if not well explained	Ignoring team-level readiness while redesigning higher-level flow
D	Regulatory response time and auditability must improve quickly	Honest capacity planning, visible compliance work, reduced last-minute escalation	Can crowd out product capacity if not governed tightly and time-boxed	Letting the overlay become a permanent parallel delivery system

How to choose between them

Do not choose based on personal preference or certification history. Choose based on enterprise needs, political constraints, and change absorption capacity. A strong practitioner can explain why the chosen model is not perfect, what it protects, and what local trade-offs leaders are accepting.

For example, if a sponsor demands a single group-wide story for regulators and the board, Sketch B may be politically comfortable but strategically weak. If architecture and data teams gate every release, Sketch A may look

neat on paper yet still fail unless Sketch C logic is built into the coordination system.

A mature recommendation sounds like this: "Given the need for cross-country reporting, shared platform bottlenecks, and limited appetite for central mandates, we recommend A with a C-style platform spine and a D overlay only for the regulatory programme through Q2."

6. Six needs scoring framework

Leaders struggle to compare operating model options because each option sounds plausible. The six needs framework gives them a decision device. Score every sketch against the needs that matter in the next twelve to eighteen months, not against theoretical perfection. Weighting matters. A bank facing a supervisory deadline should not weight local autonomy equally with regulatory certainty.

Need	Question to ask	Typical evidence	Weight guidance
1. Cross-entity delivery	How often do outcomes depend on more than one BU, country, or shared service?	Shared initiatives, missed hand-offs, duplicated planning forums	High in federated groups delivering common capabilities
2. Local autonomy	How much variation is politically or commercially unavoidable?	P&L ownership, country regulation, market-specific products	High where local leaders own material delivery choices
3. Platform bottleneck	Do shared architecture, data, or infrastructure teams slow everyone down?	Release queues, central dependency maps, recurring technical gating	High where enabling services dominate lead time
4. Regulatory certainty	Is there a hard deadline, reporting obligation, or audit exposure that must be managed in flow?	Regulatory programme load, remediation actions, audit findings	Critical in banking, utilities, and heavily supervised industries
5. Leadership enablement	Do leaders need a clear, reusable steering mechanism rather than ad hoc support?	Leadership confusion, forum sprawl, unclear escalation routes	High where sponsorship is broad but fragmented
6. Board transparency	Can the board see progress, risk, and adoption in one coherent narrative?	Conflicting dashboards, overreliance on training metrics, weak outcome reporting	High where transformation funding is under scrutiny

Scoring method

1. Agree the six needs with the sponsor group.
2. Assign each need a weight from 1 to 5.
3. Score each operating model from 1 to 5 against every need.
4. Multiply score by weight and sum the totals.
5. Discuss not only the winner, but also the assumptions hidden in the score.

Need	Weight	Sketch A	Sketch B	Sketch C	Sketch D overlay	Notes
Cross-entity delivery					N/A	How much shared cadence and dependency visibility is needed?
Local autonomy					N/A	What must remain locally adaptable without breaking enterprise coherence?
Platform bottleneck					N/A	Does the model expose or hide shared service constraints?
Regulatory certainty						Is audit evidence, sign-off, and capacity visible inside normal delivery flow?
Leadership enablement					N/A	Will leaders understand what to steer, escalate, and protect?
Board transparency					N/A	Can progress be reported consistently across the estate?
Total						Use totals as a guide, not a

						substitute for leadership judgement.
--	--	--	--	--	--	---

FACILITATION NOTE

If a workshop reaches a tie, the tie usually reveals a political issue rather than an analytical one. Surface the issue explicitly: are leaders unwilling to standardise, or unwilling to admit that enterprise reporting and compliance need more discipline than local teams currently accept?

7. Framework mix: Scrum, SAFe, Flight Levels with goal mapping table

Frameworks are useful when assigned to the correct problem. Scrum is strong at team-level accountabilities and learning loops. SAFe is useful when you need an explicit programme backbone with recurring planning and dependency management. Flight Levels is powerful when leaders need flow visibility across organisational boundaries without pretending one framework must answer every question.

Your job is not to become a framework missionary. Your job is to select the minimum mix that addresses the current constraint and to explain the choice in business terms.

Framework or concept	Best used at	What it contributes	Enterprise goal supported	Watch-out
Scrum	Team level	Clear accountabilities, cadence, backlog discipline, inspect-and-adapt loops	Raise baseline delivery capability within teams	Does not solve cross-enterprise governance or dependency management by itself
SAFe	Programme and portfolio backbone	ART structures, PI planning, RTE role, synchronised planning and review	Coordinate multiple teams on shared outcomes	Can turn into ceremonial overhead if applied without operating model choices
Flight Levels	Leadership and flow management	Visualises work across strategy, coordination, and delivery layers	Expose systemic bottlenecks and steer above team boundaries	Feels too abstract if leaders expect a full delivery operating system from it alone
Kanban discipline	Any level	WIP visibility, flow metrics, policy clarity	Improve predictability and reduce hidden queues	Easy to under-specify ownership and cadences

Goal mapping table

Enterprise goal	Scrum	SAFe	Flight Levels	Recommended use
Improve team ownership and delivery discipline	Primary	Supporting	Optional	Start with strong team accountabilities and quality discipline
Coordinate ten teams on one cross-BU outcome	Supporting	Primary	Supporting	Use a programme backbone and a shared planning cadence
Expose blocked work across portfolio, architecture, and delivery	Limited	Supporting	Primary	Use flow visualisation to make leadership constraints visible
Embed compliance and audit visibility in planning	Supporting	Primary	Supporting	Combine explicit planning cadence with visible governance queues

A credible answer to "Which framework should we adopt?" is: "Which enterprise constraint are we trying to solve first, and what is the lightest combination of practices that will solve it without buying more process than the organisation can absorb?"

8. Red flags vs strong language table

Enterprise roles are won and lost through language. Weak language signals that you are still thinking like a team coach. Strong language shows that you understand scale, governance, and leadership choices. The table below can be used as deliberate rehearsal material.

Red flag language	Why it is weak	Strong language
"I will coach the teams to be more Agile."	Frames the challenge as local behaviour change rather than system design.	"I will design the operating model and enablement path so teams can deliver within a clear enterprise backbone."
"We should roll out SAFe."	Jumps to a tool without explaining the organisational problem it addresses.	"We may need a programme backbone such as SAFe once we decide the target operating model and the constraints it must handle."
"Training should be the first step."	Creates activity before design, usually leading to low adoption and wasted spend.	"Diagnosis and operating model selection come first; training is then targeted to the roles and forums the model requires."
"The teams need to collaborate more."	Vague, non-actionable, and easy for leaders to ignore.	"The current model hides cross-BU dependencies and creates unowned waits at architecture and compliance gates."
"Our maturity score improved."	Abstract and often unconnected to business outcomes.	"Adoption rose from two to five in-scope ARTs operating to the target model, and PI predictability improved from 54% to 71%."
"We need more buy-in."	Usually means the mandate, trade-offs, or decision rights were never made explicit.	"We need a sponsor-backed decision on what is mandatory centrally and what remains locally configurable."

How to train better language

- Replace abstract nouns such as *mindset* and *maturity* with visible system properties such as flow, decision rights, and adoption.
- Anchor every recommendation in an enterprise problem: cross-entity delivery, predictability, regulatory certainty, or cost of delay.
- Make the next decision obvious. Leaders are more persuaded by "here is the decision you need to make" than by a conceptual lecture.

9. Stakeholder map: Sponsor, BU lead, EA, Compliance, HR, Portfolio — what each needs to hear

Enterprise enablement fails when practitioners speak to every stakeholder in the same voice. Different actors care about different risks. The sponsor cares about trajectory and decision quality. The enterprise architect cares about integration and sequencing. Compliance cares about evidence and accountability. HR cares about roles and manager expectations. A single generic "Agile transformation" message will land weakly with all of them.

Stakeholder	What they worry about	What they need to hear	What you should ask for
Sponsor	Board credibility, investment value, pace without chaos	"We are designing a delivery operating model with explicit adoption measures and decision points, not buying generic training."	Mandate, workshop attendance, and explicit decisions on target model plus pilot
BU lead	Loss of autonomy, central bureaucracy, delivery disruption	"The aim is to standardise only what cross-BU delivery requires and leave the rest local by design."	Visibility into dependencies, local constraints, and named representatives for model design
Enterprise architect	Architecture debt, sequencing, unrealistic delivery promises	"The operating model will align planning cadence and governance forums so architecture becomes part of flow rather than an end-stage gate."	Architecture participation in design workshops and a visible dependency management mechanism
Compliance / risk	Bypassed controls, weak evidence, late surprises	"We will reserve explicit capacity and create auditable work items inside the normal backlog flow, not in a shadow process."	Joint definition of regulatory intake, sign-off points, and evidence expectations
HR	Role ambiguity, inconsistent manager expectations, training requests without role clarity	"Role paths will be tied to operating model needs, with clear expectations for leaders, Product Owners, RTEs, and internal coaches."	Support for role descriptions, manager briefings, and capability pathways
Portfolio office	Planning integrity, reporting consistency, change overload	"The transformation will improve how work is planned and reported, not create an extra reporting layer beside the portfolio process."	Joint design of cadence, portfolio sync, and enterprise dashboard definitions

WHAT NOT TO DO

Reuse the same deck for every stakeholder

A sponsor deck should surface decisions and trade-offs. A compliance session should surface control points and evidence flows. A BU session should surface local constraints and what remains configurable.

WHAT TO DO INSTEAD

Translate one model into six stakeholder views

Hold the underlying operating model constant, but adapt the emphasis, examples, and asks so each function can see its place in the system.

USEFUL HEURISTIC

If a stakeholder conversation ends with "sounds interesting" but not with a concrete action, the message was too generic. Each stakeholder discussion should close with one decision, one owner, and one next forum.

Executives often accept the logic of enterprise Agile only when they can see how it behaved in a real environment. These cases are intentionally generic in employer naming, but concrete in structure, constraints, and signals. Their purpose is not brand association. Their purpose is to show how the operating-model-first approach behaves in practice.

10. Case A: tier-1 European bank, multi-country liquidity reporting

Situation. The assignment sat within a tier-1 European bank operating across Belgium, Hungary, the Czech Republic, Bulgaria, and India, with more than fifty onshore and offshore contributors involved in liquidity reporting and adjacent delivery streams. The programme faced a familiar tension: strong regulatory pressure, distributed teams, shared data and platform dependencies, and uneven local maturity in delivery practices.

Several delivery issues presented as team problems but were in fact system problems. Planning rhythms differed by location. Dependencies on data and risk stakeholders surfaced late. Compliance work existed partly inside the delivery system and partly outside it. Leadership saw effort and urgency, but not a clear picture of how the cross-country mechanism was supposed to function end to end.

Approach taken

Workstream	What was done	Why it mattered
Federation mapping	Mapped locations, hand-offs, shared services, key forums, and decision rights for reporting change and related data dependencies.	Made the real operating context visible instead of assuming one country or one function represented the whole system.
Programme backbone	Introduced a clearer cadence for multi-team planning and review, using programme-level coordination rather than leaving every location to self-synchronise.	Reduced late dependency discovery and created a repeatable structure for leadership attention.
Governance in flow	Worked with risk and reporting stakeholders to make compliance and reporting obligations visible in the normal work system.	Reduced shadow work and improved auditability of delivery choices.
Leadership translation	Shifted reporting from local activity metrics to cross-country signals: bottlenecks, readiness, capacity contention, and delivery confidence.	Helped sponsors make decisions on sequencing, support, and escalation rather than simply observing activity.

Signals that the approach was working

- Cross-location conversations moved from "our team is blocked" to "this dependency class recurs and needs a shared mechanism".
- Programme-level planning became more credible because upstream data and compliance work were visible sooner.
- Leadership began asking better questions about constraint removal and capacity protection instead of simply asking whether ceremonies were occurring.

- The system became explainable to new participants, which is an early indicator that the operating model is becoming institutional rather than personal.

Lessons for practitioners

The case reinforces three lessons. First, multi-country coordination requires more than stronger Scrum practice. Second, regulated work needs explicit capacity and evidence design. Third, leadership confidence increases when the operating mechanism is visible, even before all performance metrics have improved.

The breakthrough was not a better stand-up. The breakthrough was making it possible for leaders and delivery teams to see the same cross-border system and act on it with a common cadence.

11. Case B: greenfield enterprise Agile CoE in a consulting-led environment

Situation. The second case involved standing up an enterprise Agile CoE from the ground up inside a consulting-led environment with a clear mandate to move beyond one-off coaching engagements. There was appetite, sponsorship, and organisational energy, but no institutional machinery. Agile capabilities existed in pockets, largely dependent on individual consultants or local team leads.

The opportunity was ideal for a greenfield design because the mandate was broad enough to create a system, not merely a set of workshops. The risk, however, was familiar: build a services catalogue with strong branding but no authority, no role pathways, and no measurable adoption logic.

Approach taken

MANDATE

Write the CoE charter before the curriculum

The first priority was to define authority, interfaces, operating rhythms, and sponsor expectations. Training content was deliberately sequenced after the charter, not before.

CAPABILITY

Build role pathways, not generic awareness

Distinct enablement journeys were created for leaders, Product Owners, delivery leads, and internal coaches, each anchored to real operating model responsibilities.

CADENCE

Establish reusable forums

Regular steering, practice, and review rhythms turned the CoE from a training provider into a standing operating mechanism.

SCALE

Pilot services as a system

Instead of piloting isolated training, the CoE piloted an end-to-end support model: diagnostics, coaching, role enablement, and adoption reporting together.

Design choice	Rationale	Outcome signal
Written charter with sponsor backing	Prevents the CoE from becoming a voluntary community with no decision rights	Stakeholders knew when to involve the CoE and what standards it could set
Internal coach network	Reduces dependency on external delivery and creates a durable internal capability base	Knowledge transfer became part of the model rather than an end-of-project afterthought
Role-based enablement	Increases relevance and drives behavioural change at the actual points of responsibility	Leaders engaged more seriously because the content was tied to real decisions and forums
Adoption dashboard	Creates evidence of whether the model is taking hold beyond attendance numbers	Leadership could separate progress from theatre

Lessons for practitioners

A greenfield CoE is not easier than a bank programme; it merely has different traps. The absence of legacy structures does not remove the need for explicit design. If anything, the lack of constraints can encourage vague ambition. The discipline is to specify the mandate, service model, and transition path clearly enough that the CoE becomes part of the organisation's operating fabric rather than a temporary initiative.

WHAT TO COPY FROM THIS CASE

Start with the charter, then the role paths, then the cadence model, then the learning assets. In that order, the CoE becomes an operating system. In reverse order, it usually becomes a training catalogue.

12. Failure modes: fake agile, waterfall in sprints, compliance bypass, training theatre, local reinvention

Every enterprise transformation encounters the same family of failure modes. Recognising them early is one of the most valuable skills a practitioner can build. Many of these patterns look acceptable for a few months because the organisation can still generate visible activity. The problem is that the activity does not compound into enterprise capability.

Failure mode	Signals	Typical cause	Response
Fake agile	Ceremonies occur, but decision rights, funding logic, and escalation paths remain unchanged	Agile introduced as vocabulary rather than operating model	Reset the transformation around explicit enterprise design choices
Waterfall in sprints	Scope remains fixed, planning happens elsewhere, and sprints merely host progress reporting	Team rituals changed, but planning and governance stayed stage-gated	Redesign the programme backbone and planning cadence
Compliance bypass	Risk and regulatory work are managed on spreadsheets or side forums outside the backlog flow	Delivery and control functions were never integrated	Create visible intake, capacity allocation, and evidence paths inside the model
Training theatre	High attendance, many certificates, little behavioural or structural change	Training was funded before roles, forums, and standards were defined	Shift to role-based enablement tied to actual operating mechanisms
Local reinvention	Each BU invents its own scaling approach and reporting language	No mandatory backbone or weak sponsorship of central standards	Define what is centrally mandated and what is intentionally local

Why failure modes persist

They persist because each one offers short-term political comfort. Fake Agile lets leaders claim progress without changing governance. Waterfall in sprints preserves existing power structures. Compliance bypass protects schedule optics until it creates a bigger problem. Training theatre creates visible action and easy procurement. Local reinvention allows every leader to retain control. The practitioner's role is to make the longer-term cost visible while there is still time to correct course.

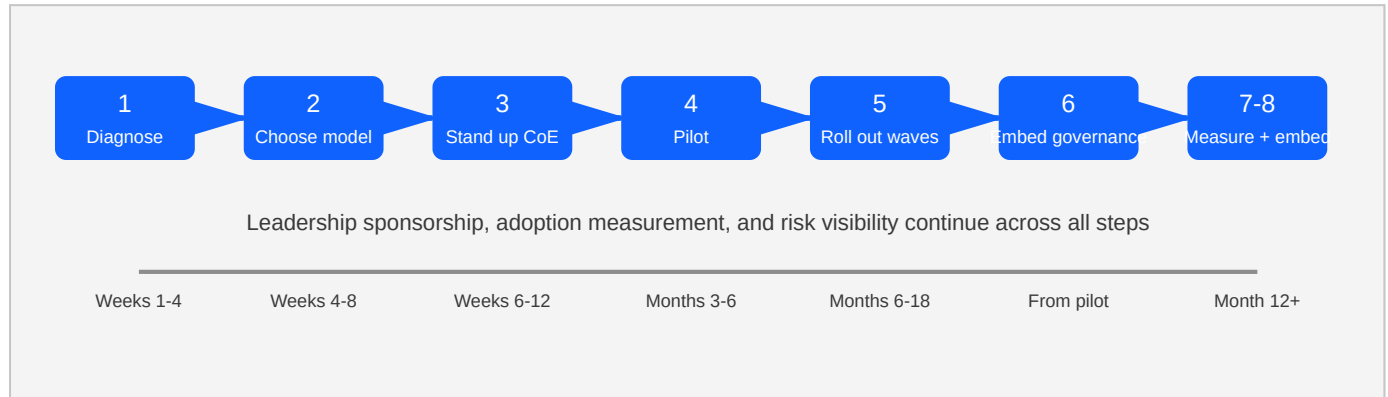
Leadership Playbook

Implementation sequence, scripts, and operating discipline

Part III translates the study material into execution. It assumes you have sponsor access, a rough mandate, and a need to move from diagnosis into implementation without becoming captive to framework theatre or endless assessment loops.

13. Eight-step rollout table with timeline and owners

The eight-step rollout is not a maturity model. It is a pragmatic sequence for moving from enterprise ambiguity to institutional capability. Some steps will overlap, but the order matters. If you skip operating model decisions, later enablement becomes confused. If you delay measurement until the end, you will not know whether adoption is real.



Step	Purpose	Indicative timing	Primary owner	Key output
1. Diagnose	Understand current state, constraints, and success criteria	Weeks 1-4	Transformation lead with sponsor	Federation map and problem statement
2. Choose operating model	Select target model and fallback before major rollout spend	Weeks 4-8	Sponsor with CoE, EA, and portfolio office	Decision record and model sketch
3. Stand up enablement	Create CoE, role paths, and rhythms that support the model	Weeks 6-12	Agile CoE lead	CoE charter and role pathways
4. Pilot	Test the model in one real federated slice	Months 3-6	Programme lead and CoE	Pilot charter and initial metrics
5. Roll out in waves	Extend to adjacent areas with entry and exit criteria	Months 6-18	Transformation lead and BU sponsors	Wave roadmap and adoption dashboard
6. Embed governance in flow	Integrate architecture, risk, and compliance with the cadence	From pilot onwards	CoE, compliance, and EA	Governance map and evidence path
7. Measure adoption	Track whether the model is actually being used	From pilot onwards	CoE and portfolio office	Adoption, flow, and predictability measures
8. Institutionalise	Transition to durable internal ownership	Month 12+	Internal CoE	Annual refresh and reduced external dependency

14. Step 1 Diagnose — detailed checklist and outputs

Diagnosis is not a survey and it is not a certification inventory. It is a disciplined attempt to understand how work actually moves through the enterprise, where it waits, which functions own the waits, and what leadership defines as success. Weak diagnosis leads to generic rollouts. Strong diagnosis creates the conditions for a legitimate operating model decision.

Diagnosis checklist

- ☐ Map business units, countries, shared services, and major value streams.
- ☐ Identify who decides capacity, prioritisation, architecture, risk, and funding.
- ☐ Observe live delivery forums rather than relying solely on stakeholder testimony.
- ☐ Capture where work waits: data teams, architecture review, procurement, compliance, release management, or leadership approval.
- ☐ Review current metrics and identify which ones are vanity indicators.
- ☐ Document whether regulatory or audit work is visible inside normal delivery flow.
- ☐ Assess role clarity for Product Owners, delivery leads, people managers, and any existing RTE equivalents.
- ☐ Agree what success means in twelve months: flow, predictability, adoption, risk reduction, or time-to-change.

Suggested interviews

Interviewee	What to learn	Question examples
Sponsor	Intent, urgency, and decision appetite	"What would make this effort credible in a year?" "What can you mandate?"
BU lead	Local constraints and autonomy boundaries	"Where do group processes help you, and where do they slow you down?"
Enterprise architect	Dependency patterns and architecture bottlenecks	"Where do delivery promises typically fail against technical reality?"
Compliance or risk	Control obligations and evidence pain	"Which delivery changes create concern because evidence arrives too late?"
Delivery leads	Ground truth on work flow and bottlenecks	"What blocks work most often, and who can resolve it?"
HR or capability lead	Role pathway realism and manager expectations	"Which roles are misunderstood or overloaded today?"

Outputs from diagnosis

OUTPUT 1

Federation map

A one-page view of the organisation showing value streams, decision hubs, and the shared services that materially affect flow.

OUTPUT 2

Problem statement

A short narrative that distinguishes systemic constraints from local symptoms and frames the next enterprise decision.

OUTPUT 3

Success criteria

Three to five measurable outcomes that leadership will use to judge whether the effort is worth continuing.

OUTPUT 4

Design inputs

Explicit constraints that any operating model must respect: regulatory, political, architectural, geographic, or capability-based.

DIAGNOSIS FAILURE PATTERN

If your diagnosis document could be reused for any company with only the logo changed, it is too generic to support an operating model decision.

15. Step 2 Operating model — sketch questions, workshop prep

The selection workshop is where ambiguity becomes a decision. To prepare properly, you need more than slide polish. You need options that are distinct enough to force a trade-off, evidence that connects those options to enterprise needs, and stakeholder preparation that prevents the room collapsing into framework preference.

Questions each sketch must answer

- What is the primary scaling unit: ART, cluster, capability group, or another construct?
- What cadence binds planning, review, and escalation together?
- Which standards are mandatory at enterprise level, and which remain local?
- Where do architecture, compliance, and portfolio governance enter the normal flow?
- What roles are essential, and what is the lightest viable role design for wave one?
- How will adoption be measured, and what would count as evidence of non-adoption?

Workshop preparation checklist

- ☐ Pre-brief the sponsor on the decision required and the likely trade-offs.
- ☐ Send diagnosis summary in advance, not the full background pack.
- ☐ Define the six needs and proposed weightings ahead of the session.
- ☐ Prepare no more than four sketches; too many options create false complexity.
- ☐ Agree who has input and who makes the final call.
- ☐ Reserve time for mandated versus local discussions; this is where hidden politics often surface.
- ☐ Plan how the decision will be documented and communicated within twenty-four hours.

Workshop agenda structure

Segment	Time	Purpose
Open with diagnosis and success criteria	30 min	Align on the problem before discussing solutions
Present model sketches	60 min	Give each option a fair explanation and explicit trade-offs
Score against six needs	75 min	Create a shared basis for comparison
Discuss mandatory vs local	30 min	Surface political boundaries before the decision is finalised
Select target model and fallback	30 min	Secure the decision and record the rationale
Close with pilot criteria and owners	15 min	Convert the model choice into immediate next actions

The workshop is not a brainstorming session. It is a decision forum. If the room leaves feeling inspired but unable to state what was chosen, the session failed.

16. Steps 3-8 summaries with bullet actions

Once the operating model is chosen, the work shifts from design to disciplined implementation. The remaining steps require fewer conceptual debates and more operational clarity. The summary below is intentionally action-oriented.

Step 3 — Stand up the enablement system

- Publish the CoE charter with sponsor endorsement.
- Define role pathways for leaders, Product Owners, RTEs, delivery leads, and internal coaches.
- Set recurring forums: leadership review, practice sync, metrics review, and escalation route.
- Define the initial tooling, templates, and artefacts needed for wave one.

Step 4 — Pilot in one real federated slice

- Select a slice with meaningful dependencies; avoid a too-easy pilot.
- Baseline current flow, predictability, and governance pain before launch.
- Run at least one full planning and review cycle under the target model.
- Capture lessons in a form that is reusable for wave two.

Step 5 — Roll out in waves

- Create entry criteria for units joining the rollout.
- Sequence waves by dependency value and readiness, not only by seniority or politics.
- Use each wave to improve the operating assets rather than merely duplicating them.
- Maintain a visible exceptions list and decide whether each exception is temporary or policy.

Step 6 — Embed governance in flow

- Integrate risk, architecture, and compliance conversations into the normal cadence.
- Define required evidence and where it is produced in delivery.
- Reserve explicit capacity for regulatory demand where needed.
- Reduce duplicate reporting by aligning governance and delivery artefacts.

Step 7 — Measure adoption

- Track adoption at the operating model level, not only by training attendance.
- Use team, programme, and enterprise metrics together.
- Review metric definitions centrally to avoid local reinterpretation.
- Escalate systemic barriers, not just lagging teams.

Step 8 — Institutionalise

- Transfer ownership of rhythms, standards, and enablement to internal roles.
- Set an annual operating model review cadence.
- Retire outdated artefacts and duplicated governance that no longer serve the model.
- Measure whether dependency on external support is falling.

SEQUENCE DISCIPLINE

The hardest part of steps 3-8 is not understanding them. It is maintaining discipline when the organisation gets impatient and asks to skip ahead. Your job is to protect the sequence while staying pragmatic about overlap.

17. Leadership scripts with objection handling

The following scripts are not theatre. They are practical structures for difficult conversations where the practitioner must remain calm, specific, and decision-oriented. Adapt the wording to your context, but preserve the logic.

Script 1 — Sponsor conversation

You: "Before we discuss frameworks or training, I want to align on what the enterprise needs from this effort in twelve months: improved flow across units, better predictability, stronger regulatory confidence, or all three."

You: "My recommendation is a short diagnosis followed by an operating model decision workshop. That gives you a target model, a fallback, and a basis for pilot investment."

Sponsor objection: "Can we not start training now to save time?"

You: "We can prepare materials, but if we train before the model is chosen we will teach roles and practices that may not fit the final design. That creates cost now and rework later."

Script 2 — Resistant BU lead

BU lead: "We already have our own way of working. A central model will slow us down."

You: "I am not proposing to standardise everything. I am proposing to standardise only what cross-BU delivery, shared services, and enterprise reporting make unavoidable. We will define explicitly what remains local."

BU lead objection: "We cannot absorb extra overhead."

You: "Then the question becomes which current overhead we can remove. A good operating model replaces duplicated forums and hidden coordination effort rather than simply adding process."

Script 3 — Compliance or risk conversation

You: "The goal is not to weaken control. The goal is to make control visible within delivery so that evidence is generated during the work, not reconstructed after it."

Compliance objection: "Agile often means people skip documentation."

You: "That happens when compliance is treated as an external gate. We are proposing explicit intake, capacity allocation, and sign-off points in the operating model, so the evidence path becomes stronger rather than weaker."

Script 4 — "Why not just train SAFe?"

Stakeholder: "Would it not be faster to put everyone through SAFe and standardise that way?"

You: "Training can accelerate adoption only after the target model is chosen. Otherwise we will certify people into a shape the organisation has not yet decided to use."

You: "If SAFe is the right backbone, we will use it. But the decision should be explained by the enterprise constraints it solves, not by the availability of a training budget."

Objection handling principles

- Do not become defensive about frameworks; return to enterprise problems and decisions.
- Validate local concerns while protecting the integrity of the operating model.
- Translate risk into system design, not reassurance alone.
- Close every difficult conversation with a next step, not a philosophical agreement.

18. SteerCo narrative — four questions leadership asks

SteerCo quality is a direct test of whether the transformation has become operationally real. Weak SteerCos review activity. Strong SteerCos answer four questions that matter to leaders and convert those answers into decisions and protections.

QUESTION 1

Are we delivering the outcomes we said mattered?

Show progress against business or regulatory outcomes, not only process adoption. Link any slippage to real constraints and decisions required.

QUESTION 2

Where is flow blocked across the enterprise?

Highlight dependency classes, wait states, and ownership gaps rather than listing isolated team impediments.

QUESTION 3

Is risk and compliance visible in the work system?

Show reserved capacity, upcoming obligations, and any evidence gaps before they become an audit surprise.

QUESTION 4

Is adoption real and repeatable?

Report how many in-scope units are operating to the target model, where exceptions remain, and whether internal capability is growing.

Recommended SteerCo slide sequence

1. **Executive summary:** progress, top risk, one decision required.
2. **Outcome view:** business and regulatory progress against the agreed success criteria.
3. **Flow and dependencies:** top bottlenecks, days open, and escalation owner.
4. **Adoption and capability:** in-scope units, role readiness, internal coaching coverage.
5. **Decisions and actions:** explicit asks, owners, and due dates.

STEERCO SMELL TEST

If a SteerCo can run without any decision being made or any blocker owner being named, it is probably operating as reporting theatre rather than a governance mechanism.

19. Metrics dictionary — team, programme, enterprise level with definitions

Metrics only help when they are defined consistently and connected to decisions. The dictionary below gives a practical baseline. You do not need all metrics immediately, but you do need clarity on which questions each metric is meant to answer.

Level	Metric	Definition	Why it matters
Team	Cycle time	Elapsed time from work starting to work meeting the agreed done criteria	Shows whether local flow is improving or masked by planning optimism
Team	Throughput	Count of completed work items per time period, interpreted with item size awareness	Useful for capacity discussions and forecasting trends
Team	Defect escape trend	Rate at which quality issues leave the team and re-enter later stages	Protects against velocity gains that erode quality
Programme	PI predictability	Percentage of planned objectives achieved within the PI, with agreed counting rules	Indicates realism of planning and coordination health
Programme	Dependency resolution days	Average time critical cross-team dependencies remain unresolved	Shows whether systemic blockers are being removed or merely observed
Programme	Release readiness confidence	Combined view of scope, quality, architecture, and compliance readiness	Creates an integrated discussion rather than fragmented status reporting
Enterprise	Operating model adoption	Share of in-scope units meeting the defined minimum conditions of the target model	Separates real scaling from training completion or self-reported maturity
Enterprise	Internal enablement coverage	Proportion of critical roles supported by internal capability rather than external dependency	Measures whether the transformation is becoming durable
Enterprise	Regulatory response visibility	Extent to which regulatory work is visible, planned, and evidenced in the delivery system	Essential in regulated environments where hidden work creates late risk

Metrics to use cautiously

- **Training attendance:** useful as an operational metric, weak as a success metric.
- **Maturity scores:** useful only if defined behaviourally and tied to the target model.
- **Velocity:** useful locally, but dangerous as a headline metric across teams or units.

20. Business case framing — cost of delay, cost of theatre, regulatory risk

Business cases for enterprise Agile fail when they promise abstract culture benefits and avoid financial or regulatory language. A stronger case focuses on avoided waste, improved decision quality, faster resolution of shared work, and reduced exposure from invisible governance.

Frame	What it means	How to use it with leadership
Cost of delay	The value lost when cross-enterprise change takes longer than necessary because dependencies and decisions are unmanaged	Show queueing, missed release opportunities, and delayed regulatory or business outcomes
Cost of theatre	The money spent on training, tooling, and ceremonial change that does not produce operating-model adoption	Contrast high activity with low adoption to show why sequence discipline matters
Regulatory risk	The exposure created when compliance work is late, invisible, or weakly evidenced	Explain why governance in flow is safer and cheaper than end-stage remediation
Leadership time value	The opportunity cost of leaders attending fragmented status forums without decision-ready information	Show how the operating model reduces forum sprawl and improves decision quality

Business case structure

1. State the current enterprise delivery problem in one paragraph.
2. Quantify visible waste where possible: delays, duplicated governance, rework, and external dependency.
3. Explain the target operating model as a control and value mechanism, not a cultural aspiration.
4. Show phased investment linked to pilot proof points and wave entry criteria.
5. Include the cost of doing nothing or doing only training.

Leaders rarely fund "becoming more Agile". They do fund better predictability, faster time-to-change, clearer accountability, and lower regulatory exposure.

21. First 90 days week-by-week table

The first ninety days determine whether you become a system designer or just another transformation voice. The table below assumes you have enough mandate to act and need a visible sequence that balances discovery, design, and movement.

Week	Focus	Key moves	Deliverable
1	Entry and alignment	Meet sponsor, clarify mandate, request access to core forums and documents	Mandate notes and stakeholder map v0.1
2	Federation mapping	Identify BUs, countries, shared services, governance routes, and value streams	Current-state federation map
3	Observation	Attend planning, review, governance, and escalation forums	Observed constraint log
4	Diagnosis synthesis	Separate symptoms from system constraints and agree success criteria	Diagnosis summary
5	Option design	Create model sketches A-D adapted to the enterprise context	Sketch pack draft
6	Pre-briefing	Socialise options with EA, compliance, portfolio office, and key BU voices	Updated pack and anticipated objections
7	Decision workshop	Run the operating model selection workshop and secure a target model	Signed decision record
8	CoE and role design	Define enablement structure, role pathways, and first rhythms	CoE charter and role map
9	Pilot definition	Select pilot slice, owners, success criteria, and baseline metrics	Pilot charter
10	Governance design	Align architecture, compliance, and reporting expectations to the cadence	Governance-in-flow map
11	Enablement launch	Start role-based support, prepare pilot planning events, and brief leaders	Enablement plan and leader pack
12	Readiness review	Check pilot readiness, confirm measures, and close major open risks	Go/no-go review and first SteerCo narrative
13	Pilot start	Launch with visible support, clear roles, and explicit follow-up loops	Pilot in motion

WHAT NOT TO DO IN THE FIRST 90 DAYS

Do not begin with a certification wave, do not let every stakeholder redefine the problem, do not select a pilot because it is politically easy, and do not wait until week twelve to define what evidence leadership will need.

22. Year 1 success criteria

Year 1 should end with evidence that the operating model is real, not merely described. These criteria should be tailored, but the categories are broadly stable across enterprise contexts.

Category	Success criteria by end of year 1
Operating model adoption	A majority of in-scope units are operating to the agreed minimum standard, with exceptions explicitly tracked and reviewed.
Pilot to rollout transition	At least one pilot has produced reusable assets and informed subsequent waves rather than remaining an isolated success story.
Leadership governance	SteerCo and related governance forums operate on decision-ready information with visible owners for systemic blockers.
Internal capability	Critical roles and coaching capability increasingly sit with internal staff rather than external dependency.
Flow and predictability	Selected flow and predictability metrics show meaningful improvement or, at minimum, increased honesty and visibility where issues remain.
Regulatory confidence	Compliance work is visible in the system, audit evidence paths are clearer, and late-surprise volume is reduced.

YEAR 1 REALITY CHECK

Success does not mean every team is perfect. Success means the enterprise can now explain how delivery works, prove where it has been adopted, and improve it using its own internal machinery.

The final section helps you secure the role, survive early setbacks, and keep a sharp eye on the anti-patterns that can quietly absorb a year of transformation effort without producing enterprise capability.

23. Interview preparation — 15 questions you should ask + 10 you should answer

At enterprise level, the interview is also an assessment of the role itself. Strong candidates do not only answer well; they diagnose whether the organisation understands the role it is hiring for. Your questions should test mandate, seriousness, and the difference between transformation intent and transformation theatre.

Fifteen questions you should ask

1. What enterprise problem is this role expected to solve in the next twelve to eighteen months?
2. What would make the effort credible to the board or executive sponsor by the end of year one?
3. Which decisions can this role influence, and which can it actually drive?
4. What is currently mandatory across business units, and what is left to local interpretation?
5. How are architecture, risk, and compliance involved in delivery today?
6. Where do the most painful dependencies sit?
7. Has the organisation already invested in framework training, and with what results?
8. Is there an existing CoE or transformation office, and what authority does it have?
9. What counts as success beyond training completion or maturity surveys?
10. How many business units, countries, or major delivery domains are in scope for year one?
11. What would be considered a successful pilot?
12. What organisational resistance should I expect to encounter?
13. Which executive forums would this role attend or influence?
14. What internal capability is expected to exist by the time external support reduces?
15. What happened in previous transformation attempts, and why did they stall?

Ten questions you should be ready to answer

Question you may be asked	What a strong answer demonstrates
How would you start in your first ninety days?	Sequence discipline: diagnosis, operating model choice, enablement, and pilot preparation
How do you decide between frameworks?	Problem-first reasoning rather than methodology loyalty
How do you work with resistant BU leaders?	Ability to balance enterprise coherence with local autonomy
What would you measure?	Understanding of team, programme, and enterprise metrics with clear purpose
How do you prevent Agile theatre?	Operating model clarity, mandate, adoption measures, and governance in flow
How would you build an Agile CoE?	Knowledge of charter, role pathways, service model, and internal capability building
How do you work in regulated environments?	Confidence integrating compliance and evidence into normal delivery work
What would a good pilot look like?	Understanding that a pilot must test meaningful enterprise constraints, not an easy corner case
How do you know whether adoption is real?	Ability to define minimum operating conditions and exceptions clearly
What is the biggest risk in enterprise Agile?	Recognition that system ambiguity, not team ignorance, is usually the central problem

A strong candidate does not sell certainty. A strong candidate shows they know which enterprise questions must be answered before certainty is possible.

24. What to do when the pilot fails

Pilot failure is not automatically evidence that the operating model was wrong. It may indicate weak sponsorship, poor pilot selection, unrealistic role expectations, or governance that never truly entered the flow. Your first job is to diagnose the nature of the failure before either defending the model or abandoning it.

Immediate response sequence

1. Stabilise the narrative. Do not let the organisation reduce the failure to "Agile does not work here".
2. Collect facts: which assumptions failed, which roles were unclear, which dependencies dominated, which governance obligations were missed.
3. Separate model failure from implementation failure. Was the design wrong, or was the chosen design never actually implemented?
4. Review pilot suitability. Was the pilot too easy, too complex, or too politically under-supported?
5. Reset with explicit decisions. If sponsorship or mandate was weak, say so directly.

Recovery options

Observed failure	Likely root cause	Recovery move
Planning cycle occurred but decisions still happened elsewhere	Leadership did not adopt the governance backbone	Reset sponsor expectations and redesign decision forums before another pilot cycle
Teams were overwhelmed by extra process	Model introduced too much too fast or roles were under-enabled	Simplify wave one standards and strengthen targeted role support
Compliance remained outside the system	Control stakeholders were not integrated early enough	Create explicit intake, capacity reservation, and evidence mapping with compliance owners
BU rejected the model as central overreach	Local autonomy boundaries were never clarified	Redefine mandatory versus local elements and reopen the decision record if needed
No reusable learning emerged	Pilot was treated as an event rather than a system test	Document lessons as operating assets and revise wave entry criteria

DO NOT DO THIS

Do not respond to pilot difficulty by launching more generic training. If the pilot failed because the system design was weak or under-adopted, more classroom time will only deepen the theatre.

25. Anti-pattern watchlist table

Keep this table close during diagnosis, rollout, and recovery. Anti-patterns rarely announce themselves. They appear as reasonable local choices that, when repeated, destroy coherence across the enterprise.

Anti-pattern	Early signs	Why it is dangerous	Countermeasure
Agile as relabelling	Project managers renamed, same stage gates remain, no changed decision rights	Creates cynicism and produces none of the intended flow benefits	Anchor the transformation in operating model choices and visible governance changes
Framework-first rollout	Training scheduled before enterprise design decisions are made	Leads to mismatched practices, rework, and false confidence	Delay broad training until the model and role paths are explicit
Local sovereignty by drift	Every unit interprets standards differently because no central minimum exists	Destroys comparability and multiplies support effort	Document mandatory versus local elements and govern exceptions
Metrics vanity	Dashboards dominated by attendance, maturity, or story counts	Hides whether the enterprise model is working	Use metrics tied to outcomes, adoption, and systemic blockers
Governance parallelism	Compliance, architecture, or risk continue to run separate, invisible queues	Creates late surprises and conflicting priorities	Design governance into the cadence and evidence flow
Hero dependency	Everything works only when one consultant or leader is present	Capability disappears when the person leaves	Build internal role paths, reusable artefacts, and standard rhythms early
Over-piloting	The organisation keeps piloting without ever setting wave criteria	Creates endless experimentation without enterprise adoption	Define what makes a pilot complete and when rollout begins
Under-powered CoE	The CoE has activity but no authority, budget, or sponsor protection	Turns enterprise enablement into an advisory side service	Secure written mandate and leadership governance around the CoE

The real anti-pattern is not "people resist change". The real anti-pattern is asking people to change without giving the enterprise a design, a mandate, and a mechanism that makes the change survivable.

CLOSING NOTE

Enterprise Agile enablement is credible when it can answer three questions at any point in time: how work is meant to flow, who is responsible for enabling that flow, and what evidence shows the organisation is moving from local effort to enterprise capability.