GUIDE K2

TURN8

HOW TO ENHANCE AI AGENT CAPABILITY: FROM MVM TO CUSTOM MODELS AND EXPANDED AUTONOMY

AI Agent Expansion and Capability Enhancement

WHERE THIS IS USED

- Venture Studio ventures with live AI agents
- CVC portfolio companies advancing AI capability
- Corporate AI Studio deployments
- Foundry-as-a-Service engagements

AUDIENCE

- Head of AI / AI Studio Lead
- ML / Data Lead
- AI Governance / Compliance Lead
- EIR / General Manager

PHASE

Phase Four: Scale & Optimize
Section K — AI Agent Expansion & Capability Enhancement
Follows Section J (Follow-On Capital & Portfolio Value)

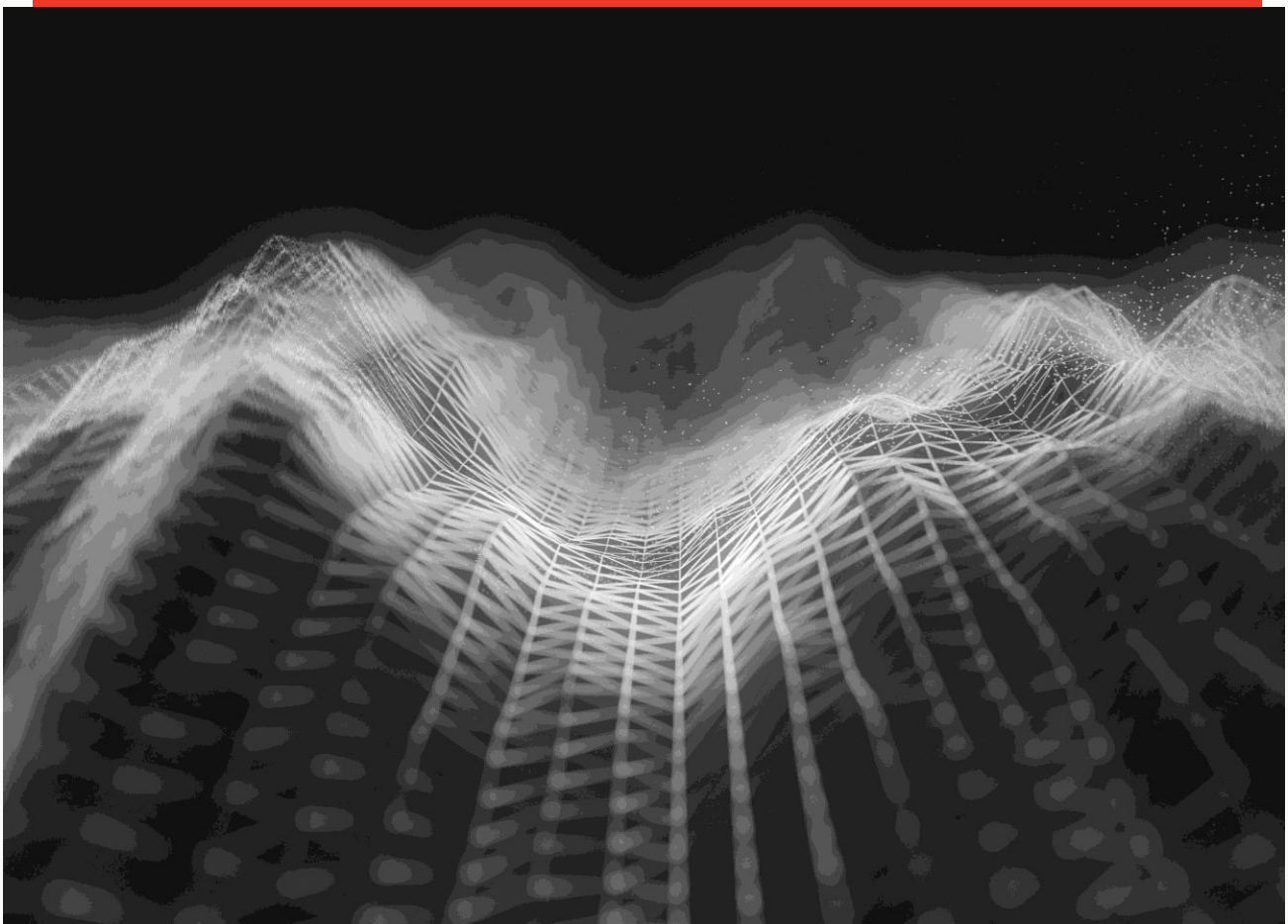
1

EXECUTIVE SUMMARY

An agent launched on the Minimum Viable Model — an off-the-shelf API with a well-designed prompt and a deliberately narrow scope — is conservative by design. Once it has run in production, accumulated proprietary data, and earned trust, the question becomes how far to advance it: whether to train a custom model, and how much autonomy to grant. Both are capability upgrades with real cost and real risk, and both must be earned rather than assumed.

This guide provides the capability roadmap that sequences those upgrades, the graduation test for when to leave the off-the-shelf API — the three-condition gate from Guide G3 applied to accumulated production data — and the guardrails for safely widening what an agent is trusted to do unsupervised.

The four outputs of this guide are: a per-agent capability roadmap; a documented custom-model decision made against the three-condition gate, with a model card and baseline evaluation if training is approved; an autonomy expansion plan with re-tuned guardrails; and a governance record authorizing each upgrade with data-localization compliance for any training data.



2

THE CORE PROBLEM

Capability enhancement is where AI ventures either compound their advantage or quietly acquire risk. The two levers — custom models and expanded autonomy — are the most attractive and the most misused. A custom model is reached for as a reflex to fix problems a better prompt would solve, and autonomy is treated as a dial to turn up rather than a boundary to govern.

A custom model is not a substitute for a well-designed system, and autonomy is not a feature. Both are governed investments that must clear a gate and carry guardrails, or they trade a small known cost for a large unknown one.

The six most consistent failure patterns in capability enhancement:

- **Custom model as reflex.** A custom model is proposed to fix a problem that a better system prompt or design would solve — expensive, slow, and often no better than the off-the-shelf API.
- **Autonomy by drift.** The agent's unsupervised scope widens informally, without a decision or new guardrails, until an incident forces a rollback.
- **Capability without data.** Custom training is attempted before enough clean, labeled, compliant production data exists to justify or support it.
- **No graduation test.** There are no explicit criteria for when to leave the off-the-shelf API, so the decision is made on enthusiasm or vendor pressure.
- **Guardrail gap.** Autonomy expands but the confidence thresholds, escalation, and reversibility that made the Minimum Viable Model safe are not re-tuned for the wider scope.
- **Upgrade without governance.** Model and autonomy changes ship without a model card, audit trail, or sign-off, invisible to governance until they fail.

The real issue:

Advancing capability without the gate and the guardrails is how a trusted agent becomes an incident. The disciplines that keep this safe are not obstacles to progress — they are what allow progress to compound: fix the system before the model, clear the three-condition gate before training, and re-tune the guardrails before widening autonomy. An upgrade that skips them is not a faster path to capability; it is a slower path to a rollback.

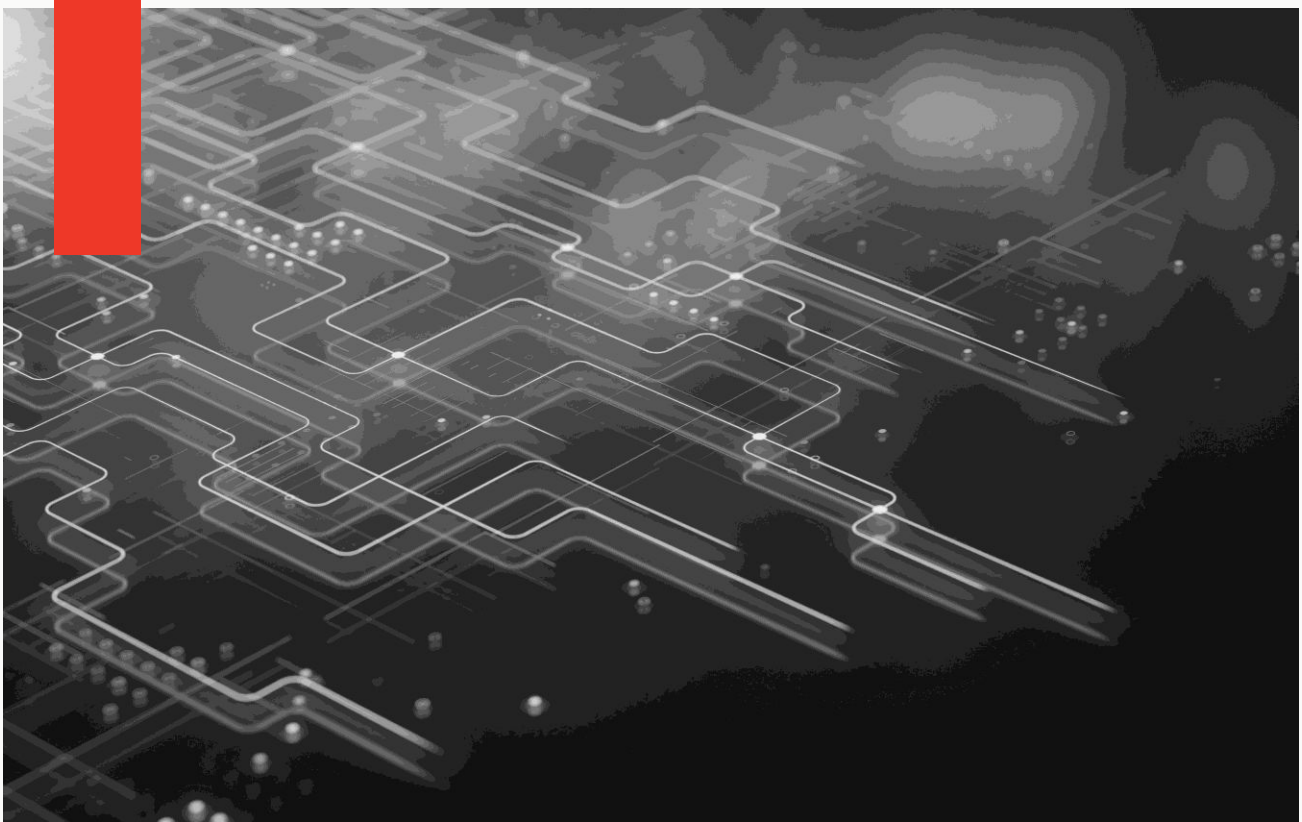
PREREQUISITES

Must Be Complete Before Starting:

- Guide G3 completed — monitoring is live, the three-condition custom-model gate is defined, and drift classification (Type 1–4) is in use
- The agent has a stable production baseline and has accumulated proprietary production data
- The Minimum Viable Model has run in production long enough to have hit, or demonstrably not hit, a quality or cost ceiling with evidence
- An AI governance process is in place for model and autonomy changes, and data-localization terms are known for any training data

Team Readiness:

- AI Studio Lead and ML / Data Lead available for the capability work
- Clean, labeled, compliant production data accessible for evaluation and any training
- AI Governance / Compliance Lead available to authorize model and autonomy changes
- For CVC tracks: the Close Owner has confirmed how capability upgrades and their value are recorded in the PMS



4

EXPECTED OUTPUT/ SUCCESS CRITERIA

You have completed this guide when the following are true:

- A per-agent capability roadmap sequencing system/prompt, model, and autonomy upgrades cheapest-reversible-first
- The three-condition graduation test applied, with a documented custom-model decision to train or not train
- If training: a model card, an evaluation against the off-the-shelf baseline, and a rollback plan
- An autonomy expansion plan that widens scope one step at a time with re-tuned guardrails
- A governance record authorizing each upgrade, with data-localization compliance confirmed for any training data
- No capability upgrade shipped without a gate cleared and guardrails set and tested

5

STEP-BY-STEP INSTRUCTIONS

STEP 1 BASELINE THE AGENT'S CURRENT CAPABILITY AND ITS CEILING

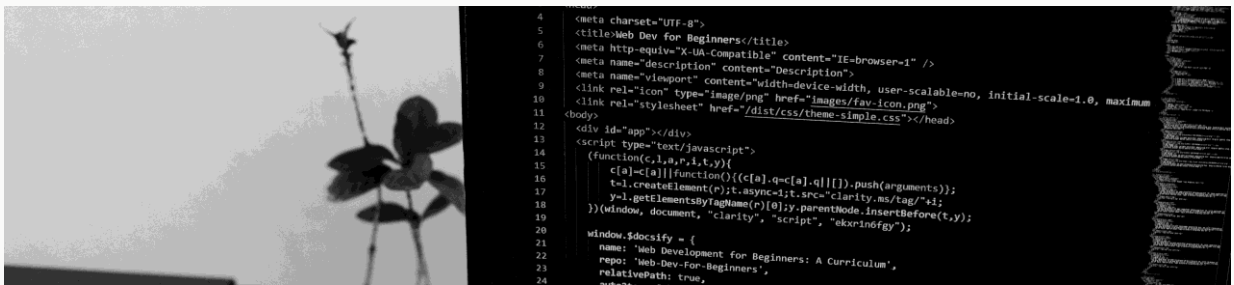
Before proposing any upgrade, establish what the Minimum Viable Model does today and where it falls short. Critically, classify the shortfall: is the gap a system or prompt problem — a Type 2 drift signal — that a better design would fix, or a genuine model-capability ceiling? Most proposed custom models are solving a Type 2 problem and should never be built.

1.1 **Document current capability against the baseline** — State what the agent does, its accuracy, latency, and cost per call against the G3 baseline, and the specific tasks where it underperforms. Precision here prevents solving a problem the agent does not actually have.

1.2 **Classify the capability gap** — Using the G3 drift classification, determine whether the shortfall is a system or prompt gap (Type 2), an input shift (Type 1), an external model change (Type 3), a workflow change (Type 4), or a true model ceiling. Only a true model ceiling justifies considering a custom model.

1.3 **Exhaust the cheap fixes first** — If the gap is a system or prompt problem, fix it there — it is faster, cheaper, and reversible. A custom model built on top of a fixable system problem inherits the problem at far greater cost.

> **AI PROMPT — Capability Baseline and Gap Classification**
>
> I am baselining an AI agent's capability before considering an upgrade. Current performance against the G3 baseline: [paste accuracy, latency p95, cost per call, and the tasks where it underperforms]. Using the G3 drift classification, help me determine whether each shortfall is a Type 2 system/prompt gap, a Type 1 input shift, a Type 3 external model change, a Type 4 workflow change, or a genuine model-capability ceiling. For each shortfall that is not a true model ceiling, propose the cheapest reversible fix. Flag clearly which shortfalls, if any, would actually justify considering a custom model.



STEP 2 BUILD THE CAPABILITY ROADMAP

Sequence the upgrades so the cheapest, most reversible improvements come first and the expensive, hard-to-reverse ones come last. The order is almost always the same: fix the system and prompt, then consider a model, then consider autonomy — because each later step is more costly and riskier than the one before.

Upgrade Tier	What It Changes	Cost / Reversibility	Consider When
System / prompt	Prompt design, tools, retrieval, workflow	Low cost, fully reversible	Any Type 2 or design gap — always first
Custom model	The underlying model, trained on your data	High cost, hard to reverse	The three-condition gate is met
Expanded autonomy	What the agent does unsupervised	Low cost, high risk	Performance and trust are proven at current scope

2.1 **Fix the system and prompt before anything else** — The first tier is always to exhaust system and prompt improvements. They are cheap, reversible, and frequently close the gap that a custom model was being proposed to solve.

2.2 **Sequence expensive, irreversible upgrades last** — Custom-model training and autonomy expansion are sequenced after the cheap fixes and only against their gates. Doing them first is how ventures spend the most to learn what a cheaper step would have told them.

STEP 3 APPLY THE GRADUATION TEST: WHEN TO LEAVE THE OFF-THE-SHELF API

Custom-model training is a Phase Four investment decision and a governance decision, not an engineering preference. Apply the three-condition gate from Guide G3: all three conditions must be confirmed before any custom-model training is proposed. If any one fails, the answer is not to train.

Condition	Confirmed When	If Not Confirmed
Sufficient proprietary data	Enough clean, labeled, compliant production data exists to train and evaluate	Do not train; keep accumulating data
Confirmed ceiling	The off-the-shelf API has demonstrably hit a quality or cost ceiling, with evidence	Do not train; the API is still improving or adequate
A model problem, not a system problem	The gap is a true model ceiling, not a Type 2 system or prompt gap	Fix the system or prompt instead

3.1 **Require all three conditions, not two of three** — The gate is a conjunction. Sufficient data plus a real ceiling still does not justify training if the problem is actually a system gap; a real model ceiling does not justify training if there is not enough data to train well.

3.2 **Record the decision as a governance decision** — Whether the answer is train or do-not-train, record it with the evidence for each condition. This is the audit trail for a Phase Four investment decision, and it protects the venture from re-litigating the same proposal.

AI PROMPT — Custom-Model Graduation Test

```
I am applying the three-condition graduation test to decide whether
to train a custom model for an agent. Evidence: (1) proprietary
training data available — [volume, quality, labeling, compliance];
(2) off-the-shelf ceiling — [the specific quality or cost ceiling
hit and the evidence]; (3) problem type — [why this is a model
ceiling and not a Type 2 system/prompt gap]. Evaluate each condition
as confirmed or not confirmed with reasoning, state whether all
three are met, and give a train or do-not-train recommendation. If
any condition fails, state exactly what would need to change before
training should be reconsidered.
```

STEP 4 IF TRAINING: RUN THE CUSTOM-MODEL BUILD WITH A MODEL CARD AND BASELINE EVALUATION

Where the gate is cleared, the custom-model build is disciplined: prepare compliant data, train, evaluate against the off-the-shelf baseline, and produce a model card before deployment. A custom model that is not measurably better than the API it replaces should not ship, regardless of the effort spent building it.

4.1 **Prepare training data with data-localization compliance** — Confirm that the training data may be used and stored where the training runs, honoring the data-localization requirements from Guide K1. Non-compliant training data is a liability baked into the model.

4.2 **Evaluate against the off-the-shelf baseline** — Evaluate the custom model on a held-out set against the off-the-shelf API on the same tasks. The custom model must beat the baseline on the quality or cost dimension that justified training, or the decision was wrong.

4.3 **Produce a model card before deployment** — Document the model's training data, intended use, performance, limitations, and evaluation in a model card before it goes anywhere near production. The model card is the governance artifact and the basis for the deployment sign-off.

AI PROMPT — Custom-Model Build and Model Card

```
I am building a custom model for an agent that has cleared the
three-condition gate. Training data: [describe source, volume,
labeling, and data-localization compliance]. The quality or cost
dimension that justified training: [state it]. Outline the build:
data preparation with compliance checks, training approach, and an
evaluation plan that compares the custom model against the off-the-
shelf baseline on the justifying dimension. Then draft the model
card sections — training data, intended use, performance versus
baseline, limitations, and evaluation — and define the rollback plan
if the model underperforms in production.
```

STEP 5 DESIGN THE AUTONOMY EXPANSION PLAN

Autonomy is the scope of what the agent does without a human in the loop. Expanding it is cheap to do and expensive to get wrong, so it is done deliberately: define current and target autonomy, widen one scope at a time, and keep every expansion reversible.

Autonomy Dimension	Current (MVM)	Target	Guardrail To Add
Decision scope	Recommends; human approves	Acts within defined limits	Confidence threshold; value cap per action
Tool use	Read-only or limited tools	Broader tool access	Scoped permissions; audit log per tool call
Exception handling	All exceptions escalate	Handles routine exceptions	Escalate above a defined risk or novelty threshold

5.1 **Widen one autonomy dimension at a time** — Expand decision scope, tool access, or exception handling one at a time, so that if performance degrades the cause is unambiguous and the change is easy to reverse.

5.2 **Keep every expansion reversible** — Each autonomy increase has a defined rollback — a way to return the agent to its previous scope immediately if a guardrail trips. Autonomy without a reverse gear is how a small error becomes a large one.

STEP 6 RE-TUNE THE GUARDRAILS FOR THE WIDER SCOPE

The guardrails that made the Minimum Viable Model safe were tuned for its narrow scope. A wider scope needs re-tuned guardrails: confidence thresholds, Human-in-the-Loop triggers, tool-use limits, and a kill switch — all tested before the autonomy is widened, not after.

6.1 **Re-set confidence thresholds and escalation triggers** — Recalibrate the confidence threshold below which the agent escalates and the conditions that trigger a Human-in-the-Loop review for the new scope. A threshold tuned for the old scope is the wrong threshold for the new one.

6.2 **Scope tool-use limits and add a kill switch** — Constrain which tools the agent may use and any per-action value or risk caps, and confirm a working kill switch that can suspend the agent instantly. Test both before widening autonomy in production.

6.3 **Test the guardrails before widening** — Run the new guardrails in shadow or a controlled test and confirm they fire correctly before the agent operates at the wider scope live. Guardrails that have never been tested are assumptions, not controls.

> **AI PROMPT — Guardrail Re-Tuning for Expanded Autonomy**
>
> ```
> I am re-tuning guardrails for an agent whose autonomy is being
> expanded. Current guardrails tuned for the MVM scope: [paste
> confidence threshold, HITL triggers, tool limits, kill switch]. The
> target autonomy expansion: [describe the wider decision scope, tool
> access, or exception handling]. Recommend re-tuned confidence
> thresholds and escalation triggers for the new scope, the tool-use
> limits and per-action caps to add, and a test plan that confirms
> each guardrail fires correctly in shadow before the autonomy is
> widened live. Identify the single failure mode the expanded autonomy
> most needs to guard against.
> ```

STEP 7 AUTHORIZE, DEPLOY, AND RECORD EACH UPGRADE

Every capability upgrade — model or autonomy — is authorized by governance, deployed with the same staged discipline as the original G1 launch, and recorded in an audit trail. The upgrade is not live until it is signed off, staged, and documented.

7.1 **Obtain governance sign-off with the model card and guardrail evidence** — Present the model card and the tested guardrails to the AI Governance Lead for written sign-off, exactly as a customer-facing agent required at first launch. No sign-off, no deployment.

7.2 **Deploy in stages with a rollback ready** — Roll out the upgrade through a shadow or staged launch as in G1, monitor against the baseline, and keep the rollback from Steps 4 and 5 ready. An upgrade that cannot be rolled back should not be deployed.

7.3 **Apply the dual-track split** — For a Venture Studio venture, the AI Studio Lead owns the upgrade and it is recorded in the venture's governance and, where proven, folded back into the K1 blueprint for the fleet. For a CVC portfolio company, the portfolio company owns the capability work while the Close Owner records the upgrade and its value in the PMS. In both tracks the three-condition gate, the guardrail discipline, and the governance sign-off are identical, and the arrangement should be reflected in the H1 Expectations Alignment Record.

AI PROMPT — Capability Upgrade Authorization Record

```
I am recording the authorization of a capability upgrade for an
AI agent. The upgrade: [custom model and/or autonomy expansion].
Evidence: three-condition gate result, model card, baseline
evaluation, re-tuned guardrails and their test results: [paste].
Draft the governance authorization record capturing what is being
deployed, the evidence supporting it, the guardrails in force,
the staged rollout and rollback plan, the named owners, and the
monitoring that will confirm it in production. Note how the
upgrade and its value should be recorded for the fleet blueprint
(K1) and, for CVC tracks, in the PMS.
```

TROUBLESHOOTING

Symptom	Likely Cause	Fix
A custom model was trained but is no better than the off-the-shelf API	The gap was a Type 2 system or prompt problem, not a model ceiling	Return to Step 1.2 and Step 3. Classify the gap correctly, fix the system or prompt, and reserve custom training for a confirmed model ceiling with all three gate conditions met.
The agent started making unsupervised errors after an update	Autonomy was widened without re-tuning the guardrails	Roll back per Step 5.2, re-tune confidence thresholds and escalation for the new scope per Step 6, and re-test before widening again.
Custom training stalled for lack of usable data	Training was attempted before the sufficient-data condition was met	Do not train yet per Step 3. Keep accumulating clean, labeled, compliant data and re-evaluate the gate when the data condition is genuinely met.
No one can say why the custom model was approved	The decision was not recorded as a governance decision	Apply Step 3.2. Record the three-condition evidence and the decision as the audit trail for a Phase Four investment.
Autonomy expanded gradually with no clear decision point	Autonomy by drift; scope widened informally	Define current and target autonomy explicitly per Step 5, widen one dimension at a time with guardrails, and authorize each change per Step 7.
A deployed model card does not exist or is incomplete	The model shipped without the governance artifact	Produce the Step 4.3 model card before any further deployment and obtain the Step 7.1 sign-off; suspend the model if it is customer-facing and unsigned.
A training dataset raised a data-residency concern	Data-localization compliance was not confirmed before training	Apply Step 4.1. Confirm the data may be used and stored where training runs, honoring the K1 data-localization controls, before resuming.

VALIDATION STEPS

Confirm each of the following before declaring the capability upgrade complete:

- Current capability is baselined and each shortfall is classified against the G3 drift types
- System and prompt fixes have been exhausted before any model or autonomy upgrade
- The capability roadmap sequences upgrades cheapest-reversible-first
- The three-condition gate has been applied and the custom-model decision recorded with evidence
- If trained: the custom model beats the off-the-shelf baseline on the justifying dimension and has a model card and rollback plan
- Training data meets data-localization requirements for where the training runs
- The autonomy expansion plan widens one dimension at a time and every expansion is reversible
- Guardrails are re-tuned for the wider scope and tested in shadow before going live
- Each upgrade has written governance sign-off and an audit trail; for CVC tracks, the PMS recording is confirmed

8

NEXT STEPS

Feed the upgraded agent into Guide K3, the continuous-optimization loop, so the new capability is measured against baseline and its financial impact is quantified rather than assumed.

Fold any proven upgrade — a prompt pattern, a guardrail configuration, or a model approach — back into the Guide K1 blueprint so the whole fleet benefits from what one agent learned.

Route the measured capability and value gains into Guide J2's strategic value case, where AI incremental value is part of what justifies the venture's place in the portfolio.

MASTER CHECKLIST

A. CAPABILITY BASELINE

- ☐ Current capability documented against the G3 baseline
- ☐ Each shortfall classified against the drift types (Type 1–4) or as a true model ceiling
- ☐ Cheap system and prompt fixes exhausted first

B. CAPABILITY ROADMAP

- ☐ Upgrades sequenced cheapest-reversible-first: system/prompt, then model, then autonomy
- ☐ System and prompt tier addressed before model or autonomy
- ☐ Expensive, irreversible upgrades sequenced last and gated

C. GRADUATION TEST

- ☐ Sufficient-proprietary-data condition evaluated with evidence
- ☐ Confirmed off-the-shelf ceiling evaluated with evidence
- ☐ Model-problem-not-system-problem condition evaluated; all three required to train
- ☐ Train or do-not-train decision recorded as a governance decision

D. CUSTOM-MODEL BUILD

- ☐ Training data prepared with data-localization compliance confirmed
- ☐ Custom model evaluated against the off-the-shelf baseline on the justifying dimension
- ☐ Model card produced before deployment; rollback plan defined

E. AUTONOMY EXPANSION

- ☐ Current and target autonomy defined per dimension
- ☐ One dimension widened at a time
- ☐ Every expansion reversible with a defined rollback

F. GUARDRAILS

- ☐ Confidence thresholds and escalation triggers re-tuned for the wider scope
- ☐ Tool-use limits and a working kill switch in place
- ☐ Guardrails tested in shadow before widening live

G. AUTHORIZATION AND RECORD

- ☐ Written governance sign-off obtained with model card and guardrail evidence
- ☐ Upgrade deployed in stages with a rollback ready
- ☐ Upgrade folded into the K1 blueprint; for CVC tracks recorded in the PMS

H. FINAL CHECK

- ☐ The system was fixed before the model; the model was gated before training
- ☐ Autonomy widened only with re-tuned, tested guardrails and a reverse gear
- ☐ Every upgrade is signed off, staged, and recorded

If the gap is a system or prompt problem
→ A custom model will not fix it. Fix the system first.