PRACTITIONER POCKET LIBRARY

The AI Delivery Prompt Pocket Library

Production-Ready AI Prompts for IT Consulting, Delivery, Agile, Engineering & Architecture

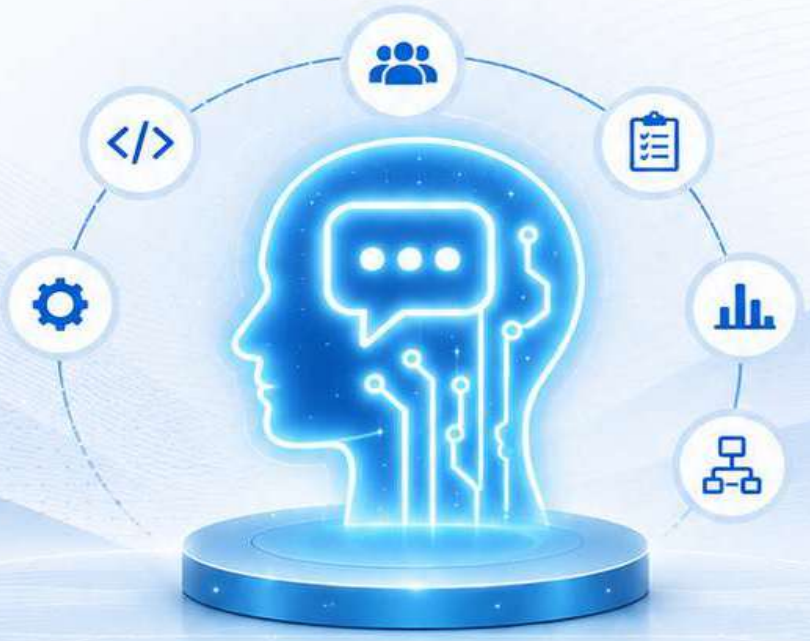
206+ Copy-Ready Prompts

19 Practitioner Roles

9 Agile Methodology Adapters

AI-Native Architecture Layer

ROLE → SITUATION → PROMPT → CONTEXT → RUN → VALIDATE → ACT



200+ PROMPTS

[DATE]

[COMPANY NAME]

[Company address]

Author

Santosh Chandankar

linkedin.com/in/santoshchandankar

chandankar.com

Version 1.0 | August 2026

PRACTITIONER POCKET LIBRARY

The AI Delivery Prompt Pocket Library

Production-Ready AI Prompts for IT Consulting, Delivery, Agile, Engineering & Architecture

206 copy-ready prompts · 19 practitioner roles · 9 Agile methodology adapters · AI-native architecture layer

ROLE → SITUATION → PROMPT → CONTEXT → RUN → VALIDATE → ACT

Version 1.0 · August 2026

Table of Contents

How to Use This Library

This is a working reference, not a tutorial. Every prompt follows the same operating sequence — find your role, match the situation, copy the prompt, fill the context, run it, validate the output, then act.

ROLE → SITUATION → PROMPT → CONTEXT → RUN → VALIDATE → ACT

- ROLE — Locate your function in the Table of Contents or the Prompt Index (Part 3).
- SITUATION — Match what's happening right now to a prompt's “Use when” line.
- PROMPT — Copy the code block exactly as written into your AI tool.
- CONTEXT — Replace every [VARIABLE] with real project data. Never leave a bracket unfilled or let the model invent one.
- RUN — Execute. Use the stated Level (L1–L4) to set the right amount of oversight.
- VALIDATE — Apply the card's “Validate” check before trusting the output.
- ACT — Follow “Next” to chain into another prompt, hand off for approval, or close the loop.

Variables

These bracketed tokens recur throughout the library. Fill each with your actual project data — do not invent values.

```
[ROLE] [PROJECT] [PROGRAM] [CUSTOMER] [BUSINESS_OUTCOME] [OBJECTIVE] [DATE] [AUDIENCE]
[CONTEXT] [SOURCE_DATA] [SYSTEMS] [CONSTRAINTS] [THRESHOLD] [RISK_TOLERANCE]
[SUCCESS_CRITERIA] [OUTPUT_FORMAT]
```

Prompt Levels

Level	Meaning
L1	Copilot / single prompt — one call, human reviews the output.
L2	Prompt chain — sequential prompts, each output feeding the next.
L3	Workflow — deterministic multi-step process with defined stages.
L4	Agent — dynamic tool use and adaptive execution within stated bounds.

Human Gate

Value	Meaning
Yes	Mandatory human sign-off before the output is used or acted on.
Conditional	Human sign-off required above a stated threshold or for a named risk category.
No	Safe for direct use once the card's Validate step has been applied.

Universal Rules (R1–R10)

- R1 — Outcome before task.
- R2 — Context before inference.
- R3 — Retrieve relevant evidence; do not assume missing information.
- R4 — Separate FACT, INFERENCE, ASSUMPTION and RECOMMENDATION when relevant.
- R5 — Use explicit IF/THEN rules for threshold-driven decisions.
- R6 — Define an exact output contract.

- R7 — Require validation before final output.
- R8 — Flag DATA GAP instead of guessing.
- R9 — Customer, financial, security, legal, production and executive outputs require human review.
- R10 — Prefer the simplest effective AI architecture.

Part 1 — The AI Delivery Prompt OS

Two reusable templates underpin every prompt in this library, plus one architecture-selection reference. Use these directly for novel situations not yet covered by a role-specific card.

Universal Prompt Template

Use for any L1–L3 prompt. Omit a component only where the use case genuinely doesn't need it — do not force every field into every prompt.

```
CONTEXT: [Engagement situation — ROLE, PROJECT, PROGRAM, CUSTOMER.]
OBJECTIVE: [The single, specific outcome this prompt must produce.]
INPUTS: [Exact data/documents/variables required — SOURCE_DATA, prior outputs, etc.]
CONSTRAINTS: [Hard limits — no invented data, no unstated assumptions, format limits.]
DECISION RULES: [Explicit IF/THEN logic for any threshold-driven judgment.]
TOOLS / SOURCES: [Specific systems, repositories or documents to ground the response — omit if
none.]
OUTPUT CONTRACT: [Exact structure/format the response must follow.]
VALIDATION: [What must be checked before the output is considered final.]
ESCALATION: [What happens on missing information (flag DATA GAP) or where human approval is
required.]
```

Universal Agent Instruction Template

Use only when an agent (L4) is genuinely justified — see R10 and the architecture reference below. Every agent instruction requires explicit stop conditions and named human-approval points.

```
GOAL: [The agent's single, bounded objective.]
CONTEXT: [Relevant project/system/business context the agent operates within.]
MEMORY: [What the agent retains across steps/sessions — and what it must not retain.]
TOOLS: [Each tool the agent may call, with an explicit scope/boundary for its use.]
DECISION RULES: [Explicit IF/THEN logic governing the agent's choices, including when to ask a
human.]
EXECUTION: [The step sequence or loop the agent follows.]
VALIDATION: [How the agent checks its own output before acting or responding.]
STOP CONDITIONS: [Explicit conditions under which the agent halts and proceeds no further.]
HUMAN APPROVAL: [Which specific actions require human sign-off before execution — see R9.]
OBSERVABILITY: [What the agent logs for every action, for audit and evaluation.]
```

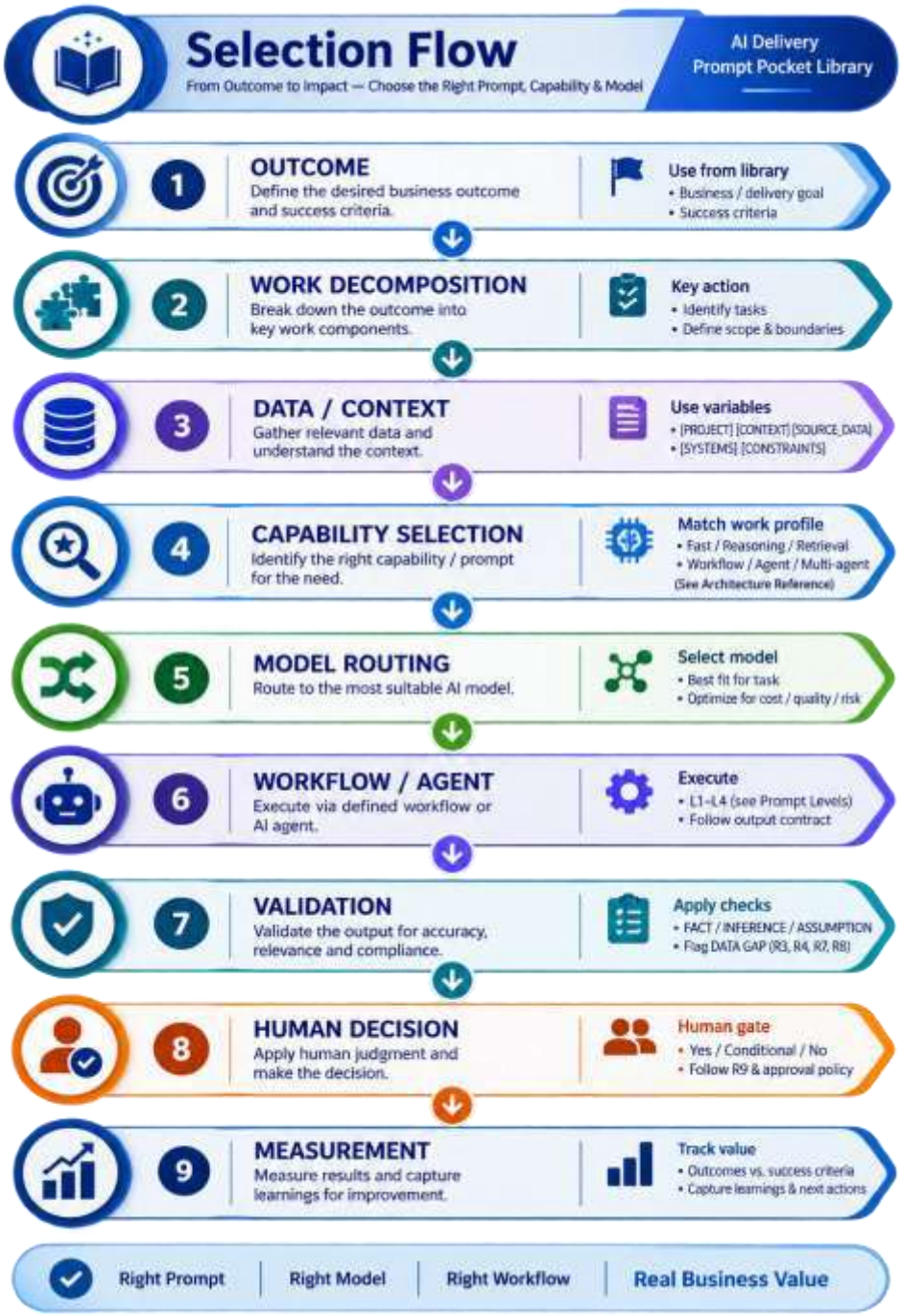
AI Architecture Quick Reference

Match the work profile to the capability before choosing a pattern. Prefer the simplest effective architecture (R10) — added complexity must be earned.

Work Profile	Preferred Capability	Optimize For
High-volume / low ambiguity	Fast model	Cost / throughput
Complex / ambiguous	Reasoning model	Quality / judgment
Knowledge-intensive	Retrieval + AI	Grounding
Deterministic	Workflow	Reliability
Dynamic	Agent	Adaptive execution
Multi-domain	Multi-agent	Specialist reasoning

Work Profile	Preferred Capability	Optimize For
Consequential	Human-gated AI	Risk / accountability

The AI Delivery Prompt Pocket Library uses a structured selection flow to move from the desired business outcome to the right AI capability, execution pattern, and measurable result. Before selecting a prompt, model, workflow or agent, practitioners decompose the work, establish the required data and context, match the work profile to the appropriate capability, validate the output, apply the required human decision gate, and measure the resulting value—ensuring AI is applied deliberately, responsibly, and with the simplest effective architecture.



Part 2 — Role-Based Prompt Library

206 prompts across 19 practitioner roles, grouped by delivery function. Every card is copy-ready — see "How to Use" for the operating sequence.

I structured this library around a simple observation from my delivery experience: **AI creates the most value when it is embedded across the delivery ecosystem, not treated as a standalone technology capability.** The 206 prompts are therefore organized around the practitioner roles that shape, execute, govern, assure, and transform technology delivery.

The distribution across **19 practitioner roles and six delivery functions** is intentional. It reflects the reality that modern technology initiatives are cross-functional—delivery leaders manage outcomes and governance, Agile and product practitioners shape requirements and priorities, engineers and architects make technical decisions, operations and quality teams protect reliability, customer-facing roles shape engagement, and AI delivery specialists connect these capabilities with emerging AI patterns.

From Individual Tasks to an Integrated Delivery Capability

The largest concentration of prompts is in **Engineering & Architecture**, followed by **Delivery & Program Leadership** and **Agile & Product**. This reflects where delivery teams encounter high volumes of analysis, decisions, dependencies, technical trade-offs, and changing information. The library also extends into **Operations & Quality**, **Customer & Engagement**, and **AI Delivery & AI Architecture**, creating a broader coverage model rather than limiting AI adoption to development or experimentation.

A practical, role-based collection of **copy-ready AI prompts** for real delivery situations across leadership, Agile, engineering, architecture, operations, customer engagement, and AI delivery. Each prompt follows the library's operating sequence:

Section	Roles	Prompts
Delivery & Program Leadership	TDM, PM, TPM, PMO	48
Agile & Product	SM, PO, BA	38
Engineering & Architecture	TL, SA, CA, IA, EA, SEC, DA	66
Operations & Quality	OPS, QA, SVC	28
Customer & Engagement	PRE	8
AI Delivery & AI Architecture	AIX	18

Delivery & Program Leadership

48 Prompts Across 4 Leadership Roles

Technology delivery rarely fails because information is unavailable; it often becomes difficult because information is fragmented across workstreams, schedules, RAID logs, technical teams, stakeholders, financial data, and governance forums. I designed this section to help delivery leaders use AI to bring that information together, identify what matters, and turn it into structured delivery intelligence and decision support.

The 48 prompts cover four roles that operate at different levels of the delivery system:

Role	Prompts	Primary Focus
Technical Delivery Manager (TDM)	15	Delivery health, risks, dependencies, critical path, capacity, technical debt, escalation, recovery and delivery intelligence
Project Manager (PM)	12	Scope, schedule, cost, risk, resources, stakeholders, governance, change and reporting
Program Manager / TPM	11	Cross-workstream coordination, program dependencies, technical delivery, milestones, risks and executive alignment
PMO / Transformation Lead	10	Governance, portfolio visibility, transformation execution, standards, performance and value tracking

Designed Around Real Delivery Decisions

The prompts are intentionally focused on situations where a delivery leader needs to understand, assess, communicate, decide, or act. Rather than asking AI to simply summarize project information, the prompts help structure questions such as:

What is changing? → Why does it matter? → What evidence supports it? → What is the delivery impact? → What decision or action is required?

This approach is particularly useful when dealing with delivery health, emerging risks, dependency cascades, milestone slippage, capacity constraints, technical debt, change requests, governance decisions, and executive escalations.

From Delivery Data to Decision Intelligence

A key design consideration is the relationship between prompts. The cards are not intended to operate only as isolated AI interactions. They can form practical delivery chains.

For example, a delivery leader can move from a health assessment to risk or dependency analysis, then into an escalation or recovery decision, and finally use the resulting insight for governance communication. Similarly, a project change can be assessed across scope, schedule, cost, and risk before it reaches the appropriate decision authority.

This makes the prompt library useful not only for productivity, but also for improving the consistency and traceability of delivery decisions.

AI as a Delivery Copilot—not the Delivery Owner

The prompts are designed to strengthen the practitioner's ability to analyse evidence, surface patterns, structure options, and prepare decision-ready outputs. The responsibility for consequential decisions remains with the appropriate delivery or business authority.

That distinction is fundamental to how I approached this section: AI can accelerate the work of a delivery leader, but accountability for delivery outcomes cannot be delegated to the model.

The result is a practical leadership toolkit that connects delivery intelligence, governance, risk management, decision support, and execution—helping practitioners spend less time assembling information and more time acting on what the information means.

Technical Delivery Manager (TDM)

Multi-workstream delivery oversight: health, risk, dependencies and executive communication.

TDM-01 Delivery Health Snapshot

Use when: You need a defensible, evidence-based read on overall delivery health before reporting up.

Outcome: A structured RAG assessment of delivery health with named drivers, not a vibe check.

Inputs: [PROJECT], [SOURCE_DATA] (status reports, RAID log, burndown/burnup, milestone tracker), [THRESHOLD] for RAG bands

PROMPT

```
CONTEXT: I am the Technical Delivery Manager for [PROJECT] under [PROGRAM] for [CUSTOMER].
OBJECTIVE: Produce a delivery health assessment across scope, schedule, cost, quality, risk and
team capacity.
INPUTS: [SOURCE_DATA] — latest status reports, RAID log, milestone tracker, burndown/burnup data.
CONSTRAINTS: Use only the data provided. Do not infer trend direction from a single data point.
DECISION RULES: RED if any dimension breaches [THRESHOLD] or has no mitigation owner. AMBER if
trending toward breach within 2 reporting cycles. GREEN otherwise.
OUTPUT CONTRACT: Table with columns [Dimension | RAG | Evidence | Driver | Trend
(Improving/Stable/Worsening) | Owner]. Follow with a 3-sentence overall narrative.
VALIDATION: Every RAG rating must cite the evidence field it is based on; no rating without
evidence.
ESCALATION: If a dimension lacks sufficient data to rate, mark "DATA GAP" instead of guessing and
list what is needed.
```

Output: RAG delivery health table + narrative summary, ready for status pack.

Validate: Every rating traces to cited evidence; no dimension rated on assumption.

Next: Feed RED/AMBER items into TDM-10 Escalation Brief or TDM-12 Steering Narrative.

LEVEL L1 HUMAN GATE No

TDM-02 Multi-Workstream Status Roll-up

Use when: You manage several workstreams and need one consolidated status view for governance.

Outcome: A single roll-up status that preserves workstream-level nuance without flattening it.

Inputs: [PROGRAM], list of workstreams with [SOURCE_DATA] status inputs, [DATE]

PROMPT

```
CONTEXT: [PROGRAM] has multiple workstreams reporting independently; I need one consolidated view
for [DATE].
OBJECTIVE: Roll up individual workstream statuses into a single program-level status without
losing material detail.
INPUTS: Per-workstream status inputs: [SOURCE_DATA].
CONSTRAINTS: Do not average RAG ratings. The program rating is the worst material rating unless
explicitly justified.
DECISION RULES: Program status = RED if any workstream on the [CUSTOMER] critical path is RED.
Otherwise AMBER if 2+ workstreams are AMBER.
OUTPUT CONTRACT: Table [Workstream | RAG | Key Progress | Key Risk | Decision Needed] plus one
program-level RAG with justification.
VALIDATION: Confirm the program-level rating logic was applied correctly, not just the highest-
count colour.
ESCALATION: Flag any workstream with stale status data (>1 reporting cycle old) as DATA GAP.
```

Output: Consolidated multi-workstream status table with program-level RAG.

Validate: Program RAG logic matches decision rules, not simple majority.

Next: Use as direct input to TDM-15 Weekly Delivery Intelligence Digest.

LEVEL L1 HUMAN GATE No

TDM-03 RAID Log Triage & Refresh

Use when: Your RAID log is stale, duplicated, or has unowned/undated items.

Outcome: A cleaned, triaged RAID log with ownership, dates and stale-item flags.

Inputs: [SOURCE_DATA] current RAID log export, [DATE]

PROMPT

```
CONTEXT: Reviewing the RAID log for [PROJECT] as of [DATE].
OBJECTIVE: Triage the existing RAID log — deduplicate, flag stale or unowned items, and recommend
closures.
INPUTS: [SOURCE_DATA] full RAID export (Risks, Assumptions, Issues, Dependencies).
CONSTRAINTS: Do not invent an owner or due date. Do not close an item without stated evidence of
resolution.
DECISION RULES: Flag STALE if no update in >14 days. Flag UNOWNED if no named owner. Recommend
CLOSE only if evidence shows resolution; otherwise recommend KEEP OPEN.
OUTPUT CONTRACT: Table [ID | Type | Statement | Status Recommendation | Reason | Owner or DATA
GAP].
VALIDATION: Every recommended closure cites the specific evidence line that justifies it.
ESCALATION: Route all UNOWNED items to the delivery lead for assignment before next steering
review.
```

Output: Triaged RAID table with close/keep/stale recommendations.

Validate: No closure recommended without cited resolution evidence.

Next: Feed unresolved risks into TDM-04 Risk Exposure Scoring.

LEVEL L1 HUMAN GATE No

TDM-04 Risk Exposure Scoring

Use when: You need to prioritize risks quantitatively rather than by gut feel.

Outcome: A scored, ranked risk register using consistent probability × impact logic.

Inputs: Risk list, [RISK_TOLERANCE], [THRESHOLD] for escalation

PROMPT

```
CONTEXT: Scoring open risks for [PROJECT] against [RISK_TOLERANCE].
OBJECTIVE: Score and rank open risks by probability × impact and identify which exceed acceptable
exposure.
INPUTS: Open risk list with available evidence for probability and impact.
CONSTRAINTS: Score probability and impact on a 1–5 scale each; do not use unscaled qualitative
labels only.
DECISION RULES: IF (probability × impact) >= [THRESHOLD] THEN escalate to steering. IF risk has
no mitigation owner AND score >= 15 THEN flag CRITICAL — ESCALATE NOW.
OUTPUT CONTRACT: Ranked table [Risk | Probability | Impact | Score | Mitigation Status | Action
Required], sorted descending by score.
VALIDATION: Recompute the top 3 scores manually to confirm the model applied the scale
consistently.
ESCALATION: CRITICAL-flagged risks require same-week TDM/sponsor review — do not hold for the
next steering cycle.
```

Output: Ranked, scored risk register with escalation flags.

Validate: Top-ranked risks manually re-checked for scoring consistency.

Next: CRITICAL items go directly to TDM-10 Escalation Brief.

LEVEL L1 HUMAN GATE Conditional

TDM-05 Dependency Cascade Analysis

Use when: A change or delay in one workstream may ripple through others and you need to trace the blast radius.

Outcome: A mapped cascade of downstream impacts from a single dependency change, with timing.

Inputs: Dependency map / RAID dependencies, the triggering event, [SYSTEMS] or workstreams affected

PROMPT

```
CONTEXT: [PROJECT] has a dependency change: <describe trigger event>. I need to understand
downstream impact.
OBJECTIVE: Trace the cascade of impacts across dependent workstreams, teams or [SYSTEMS] from
this single change.
INPUTS: Dependency register, workstream schedules, [SYSTEMS] integration map.
CONSTRAINTS: Trace only documented dependencies. Do not assume a dependency exists without it
appearing in [SOURCE_DATA].
DECISION RULES: Classify each downstream impact as DIRECT (explicit dependency) or INDIRECT
(shared resource/system, inferred — mark as INFERENCE).
OUTPUT CONTRACT: Cascade table [Downstream Item | Impact Type (DIRECT/INDIRECT) | Estimated
Delay/Effect | Confidence | Recommended Action] plus a one-paragraph narrative of the critical
chain.
VALIDATION: Every DIRECT entry must map to a specific dependency record; every INDIRECT entry
must be labeled INFERENCE.
ESCALATION: If cascade reaches a [CUSTOMER]-facing milestone, flag for TDM-13 Executive Decision
Brief.
```

Output: Cascade impact table distinguishing direct vs inferred downstream effects.

Validate: DIRECT vs INFERENCE labeling checked against dependency register.

Next: Feed critical-path items into TDM-06 Critical Path Impact Assessment.

LEVEL L2 HUMAN GATE Conditional

TDM-06 Critical Path Impact Assessment

Use when: A specific delay or scope change needs to be tested against the critical path.

Outcome: A clear verdict on whether an event moves the end date, with the driving chain shown.

Inputs: Schedule/critical path data, the specific change event, [DATE]

PROMPT

```
CONTEXT: Assessing whether <describe change/delay> affects the critical path of [PROJECT] as of
[DATE].
OBJECTIVE: Determine if the event shifts the project end date and by how much.
INPUTS: Current schedule with task dependencies and float/slack data.
CONSTRAINTS: Base the verdict only on documented float values; do not estimate float without
schedule data.
DECISION RULES: IF task float < delay duration THEN critical path shifts by (delay – float). IF
float >= delay THEN no schedule impact, absorbed by slack.
OUTPUT CONTRACT: Verdict line (Impact: Yes/No, X days) + supporting chain of tasks with float
shown + recommended response options.
VALIDATION: Confirm float values used come from the current schedule baseline, not an outdated
version.
ESCALATION: If schedule data is missing float values, flag DATA GAP and request updated schedule
before issuing a verdict.
```

Output: Critical-path impact verdict with day-count and supporting task chain.

Validate: Float figures confirmed against the current, dated schedule baseline.

Next: If impact confirmed, initiate TDM-07 Milestone Slippage Diagnosis.

LEVEL L1 HUMAN GATE No

TDM-07 Milestone Slippage Diagnosis

Use when: A milestone has slipped or is at risk and you need root cause before proposing recovery.

Outcome: A fact-based diagnosis separating root cause from contributing and symptomatic factors.

Inputs: Milestone history, [SOURCE_DATA] status notes, RAID log

PROMPT

```
CONTEXT: [PROJECT] milestone "<milestone name>" has slipped from <original date> to <current
forecast>.
OBJECTIVE: Diagnose the root cause(s) of the slippage, distinct from symptoms.
INPUTS: Milestone tracking history, status notes, related RAID entries.
CONSTRAINTS: Distinguish FACT (documented in status history), INFERENCE (reasonable read of the
evidence) and ASSUMPTION (unverified) for every cause cited.
DECISION RULES: A cause qualifies as ROOT CAUSE only if removing it would have prevented the
slip; otherwise classify as CONTRIBUTING or SYMPTOM.
OUTPUT CONTRACT: Table [Cause | Classification (Root/Contributing/Symptom) | Evidence Type
(Fact/Inference/Assumption) | Source]. Close with a 1-sentence root-cause statement.
VALIDATION: Root cause claim must be traceable to at least one FACT-classified entry.
ESCALATION: If evidence is insufficient to isolate a root cause, state DATA GAP and specify what
evidence would resolve it.
```

Output: Root-cause diagnosis table with fact/inference/assumption tagging.

Validate: Stated root cause backed by at least one FACT-tagged data point.

Next: Proceed to TDM-11 Recovery Plan Builder.

LEVEL L2 HUMAN GATE No

TDM-08 Resource & Capacity Risk Assessment

Use when: You suspect a capacity shortfall but need it quantified before raising it.

Outcome: A quantified capacity gap by role/skill and period, with risk rating.

Inputs: Resource plan / allocation data, forecast demand, [THRESHOLD] for acceptable utilization

PROMPT

```
CONTEXT: Reviewing resource capacity for [PROJECT] over the next reporting horizon.
OBJECTIVE: Quantify capacity gaps by role and time period and rate the delivery risk they create.
INPUTS: Current allocation/roster, forecast demand by role, planned leave/attrition if known.
CONSTRAINTS: Use only documented allocation percentages; do not assume unstated availability.
DECISION RULES: IF utilization > [THRESHOLD] for 2+ consecutive periods THEN HIGH risk. IF a
required skill has zero allocated FTE in a period THEN CRITICAL gap.
OUTPUT CONTRACT: Table [Role/Skill | Period | Demand | Supply | Gap | Risk Rating] + ranked list
of top 3 gaps with delivery consequence.
VALIDATION: Cross-check gap calculation arithmetic (Demand – Supply) for each row.
ESCALATION: CRITICAL gaps require resourcing decision from delivery leadership within 5 business
days.
```

Output: Capacity gap table by role/period with risk ratings.

Validate: Gap arithmetic independently re-checked per row.

Next: CRITICAL gaps feed TDM-10 Escalation Brief.

LEVEL L1 HUMAN GATE Conditional

TDM-09 Technical Debt Delivery Impact

Use when: You need to translate technical debt into delivery/schedule/cost terms for non-technical stakeholders.

Outcome: A business-framed statement of how specific technical debt items affect delivery outcomes.

Inputs: Technical debt register/backlog, upcoming roadmap items, [SYSTEMS] affected

PROMPT

```
CONTEXT: [PROJECT] carries known technical debt in [SYSTEMS] that may affect the upcoming
roadmap.
OBJECTIVE: Translate each debt item into concrete delivery impact — velocity, risk, or cost — for
governance reporting.
INPUTS: Technical debt register, upcoming roadmap/backlog items, prior incident/defect data if
available.
CONSTRAINTS: Only claim delivery impact where evidence connects the debt item to a roadmap item
or incident history.
DECISION RULES: Rate impact HIGH if debt blocks or slows a committed roadmap item; MEDIUM if it
increases defect/incident likelihood; LOW if isolated with no near-term roadmap contact.
OUTPUT CONTRACT: Table [Debt Item | Affected System | Roadmap Item Affected | Impact Rating |
Recommended Action (Fix now/Schedule/Accept)].
VALIDATION: Every HIGH rating must name the specific roadmap item it blocks.
ESCALATION: Items rated HIGH with no funded remediation plan escalate to TDM-13 Executive
Decision Brief.
```

Output: Business-framed technical debt impact table with recommended actions.

Validate: HIGH ratings tied to a named roadmap item, not general concern.

Next: Unfunded HIGH items go to TDM-13 Executive Decision Brief.

LEVEL L2 HUMAN GATE No

TDM-10 Escalation Brief Generator

Use when: An issue exceeds team-level resolution authority and needs sponsor/steering decision.

Outcome: A one-page escalation brief that gives a decision-maker everything needed to decide fast.

Inputs: The issue/risk, [SOURCE_DATA] evidence, options already considered, [DATE] decision needed by

PROMPT

```
CONTEXT: Escalating an issue on [PROJECT] that requires a decision beyond delivery-team
authority, needed by [DATE].
OBJECTIVE: Produce a one-page escalation brief enabling a fast, informed decision.
INPUTS: Issue description, supporting evidence, options already evaluated, delivery impact of
inaction.
CONSTRAINTS: State facts and options neutrally; do not pre-select the "recommended" option
without stating trade-offs for alternatives.
DECISION RULES: Include a "cost of no decision by [DATE]" line — this is mandatory, not optional.
OUTPUT CONTRACT: Sections — Situation (3 sentences) | Impact if unresolved | Options (min. 2,
each with pros/cons/cost) | Recommendation | Decision needed by [DATE] | Decision owner.
VALIDATION: Confirm at least two genuinely distinct options are presented, not one option and a
strawman.
ESCALATION: This output itself IS the escalation artifact — route directly to the named decision
owner; human decision required.
```

Output: One-page executive-ready escalation brief with options and recommendation.

Validate: At least two credible options presented with trade-offs, not a strawman.

Next: Send directly to decision owner; log decision in RAID once made.

LEVEL L1 HUMAN GATE Yes

TDM-11 Recovery Plan Builder

Use when: A milestone, budget or scope item is off track and needs a credible path back to green.

Outcome: A recovery plan with specific actions, owners, dates and expected RAG improvement.

Inputs: Root cause (from TDM-07), current status, [CONSTRAINTS] (budget, resourcing, contractual)

PROMPT

```
CONTEXT: [PROJECT] is off track on <dimension> due to <root cause from diagnosis>. Building a
recovery plan.
OBJECTIVE: Define a credible recovery path back to acceptable status within [CONSTRAINTS].
INPUTS: Root cause analysis, current baseline, known constraints (budget cap, contractual dates,
resource limits).
CONSTRAINTS: Every recovery action must address the stated root cause, not just the symptom. Do
not propose actions requiring resources known to be unavailable per [CONSTRAINTS].
DECISION RULES: Rank actions by (impact on recovery ÷ implementation effort); sequence highest-
ratio actions first.
OUTPUT CONTRACT: Table [Action | Root Cause Addressed | Owner | Target Date | Expected Impact] +
a revised forecast statement (new date/cost/scope estimate) + residual risk after recovery.
VALIDATION: Every action traces back to a specific root cause; no generic "increase focus"
actions.
ESCALATION: If recovery cannot restore acceptable status within [CONSTRAINTS], state this
explicitly and escalate scope/budget/date trade-off decision to sponsor.
```

Output: Recovery action plan with owners, dates, and a revised delivery forecast.

Validate: Each action explicitly maps to a diagnosed root cause.

Next: Present via TDM-12 Steering Narrative; track actions in RAID.

LEVEL L2 HUMAN GATE Yes

TDM-12 Steering Committee Narrative

Use when: You need to turn raw status data into a coherent narrative for a steering/governance forum.

Outcome: A steering-ready narrative that leads with decisions needed, not a status recap.

Inputs: Latest health snapshot, RAID highlights, [AUDIENCE], [DATE] of the forum

PROMPT

```
CONTEXT: Preparing the steering committee narrative for [PROJECT] / [PROGRAM] for the [DATE]
governance forum, audience: [AUDIENCE].
OBJECTIVE: Convert status data into a narrative that leads with decisions needed and only then
covers progress.
INPUTS: Delivery health snapshot, top RAID items, any pending escalations.
CONSTRAINTS: Maximum 5 narrative bullets before the decisions section. No unresolved acronyms for
a non-technical [AUDIENCE].
DECISION RULES: Anything requiring steering input must appear in a "Decisions Needed" section,
not buried in status narrative.
OUTPUT CONTRACT: Sections — Headline (1 sentence) | Decisions Needed (bulleted, with options) |
Progress Highlights (max 5 bullets) | Top Risks | Ask.
VALIDATION: Confirm every item in "Decisions Needed" has at least two options and an owner.
ESCALATION: N/A — this is a communication artifact; underlying unresolved risks still route via
TDM-10.
```

Output: Steering-ready narrative structured around decisions, not status.

Validate: Every decision item includes options and an accountable owner.

Next: Pair with TDM-13 Executive Decision Brief for any single high-stakes ask.

LEVEL L1 HUMAN GATE Yes

TDM-13 Executive Decision Brief

Use when: A single high-stakes decision needs to go to an executive/sponsor outside a regular forum.

Outcome: A concise decision brief isolating one decision, its options, and its consequences.

Inputs: The decision required, options considered, financial/risk implications, [DATE] needed by

PROMPT

```
CONTEXT: A decision is required from [CUSTOMER]/sponsor leadership on [PROJECT] by [DATE].
OBJECTIVE: Present a single decision with enough evidence for an executive to decide without
follow-up questions.
INPUTS: Decision statement, evaluated options, cost/schedule/risk implications of each.
CONSTRAINTS: One decision per brief. Do not bundle multiple asks. Quantify impact in cost,
schedule, or risk terms wherever data allows.
DECISION RULES: If quantification is not possible from available data, state DATA GAP rather than
a placeholder number.
OUTPUT CONTRACT: Sections — Decision Required (1 sentence) | Why Now | Options (each with
cost/schedule/risk impact) | Recommendation with rationale | Owner's Decision (blank for sign-
off).
VALIDATION: Confirm the brief contains exactly one decision and each option has quantified or
explicitly-flagged-as-unquantified impact.
ESCALATION: This document requires named executive sign-off before proceeding — treat as a human-
gated artifact.
```

Output: Single-decision executive brief with quantified options and sign-off field.

Validate: Exactly one decision per brief; each option's impact is quantified or flagged.

Next: Log final decision in program RAID and update TDM-01 health snapshot.

LEVEL L1 HUMAN GATE Yes

TDM-14 Vendor / Partner Performance Review

Use when: You need a structured, evidence-based review of a vendor or delivery partner.

Outcome: A scored vendor performance review against contracted commitments, not general impressions.

Inputs: Contract/SOW commitments, delivery evidence (SLA data, deliverable dates, defect rates), review period

PROMPT

```
CONTEXT: Reviewing <vendor/partner name> performance on [PROJECT] for the period ending [DATE].
OBJECTIVE: Score vendor performance against contracted commitments with supporting evidence.
INPUTS: Contract/SOW commitments (SLAs, deliverables, quality KPIs), actual delivery evidence for
the period.
CONSTRAINTS: Score only against explicitly contracted or agreed commitments; do not penalize for
uncontracted expectations.
DECISION RULES: MEETS if within contracted threshold; BELOW if outside threshold once; AT RISK if
below threshold in 2+ consecutive periods.
OUTPUT CONTRACT: Table [Commitment | Contracted Threshold | Actual | Rating | Trend] + narrative
on any AT RISK items and recommended contractual action.
VALIDATION: Every rating traces to the specific contract clause and evidence source used.
ESCALATION: AT RISK ratings route to commercial/procurement lead for contractual remedy
discussion — human review required.
```

Output: Evidence-based vendor scorecard against contracted commitments.

Validate: Ratings traceable to specific contract clauses and evidence.

Next: AT RISK items require commercial lead review before next vendor governance call.

LEVEL L2 HUMAN GATE Yes

TDM-15 Weekly Delivery Intelligence Digest

Use when: You need a fast, consistent weekly synthesis across all workstreams for your own situational awareness.

Outcome: A one-page weekly digest surfacing what changed, what's new-risk, and what needs attention.

Inputs: This week's [SOURCE_DATA] vs last week's, RAID delta, [DATE]

PROMPT

```
CONTEXT: Weekly delivery intelligence synthesis for [PROJECT]/[PROGRAM], week ending [DATE].
OBJECTIVE: Surface what materially changed this week versus last, not a full status restatement.
INPUTS: This week's status/RAID data and last week's equivalent for comparison.
CONSTRAINTS: Only report deltas (new, closed, worsened, improved items). Do not restate unchanged
items in detail.
DECISION RULES: Flag as "NEEDS ATTENTION" anything that moved from GREEN/AMBER to RED, or any new
CRITICAL risk (see TDM-04).
OUTPUT CONTRACT: Sections — New This Week | Resolved This Week | Deteriorated | Improved | Needs
Attention (with recommended owner) — each as a short bulleted list.
VALIDATION: Every item classified as a genuine week-over-week delta, not carried-over status.
ESCALATION: "Needs Attention" items with no assigned owner escalate to TDM delivery lead review.
```

Output: One-page week-over-week delivery delta digest.

Validate: Confirm each entry is an actual delta, not unchanged status restated.

Next: Use as the weekly input to TDM-01 Delivery Health Snapshot.

LEVEL **L1** HUMAN GATE **No**

Project Manager (PM)

Core PMBOK-aligned project management: scope, schedule, cost, risk, governance and reporting.

PM-01 Scope Baseline & Boundary Definition

Use when: Scope is ambiguous or contested and you need a clear, testable in/out boundary.

Outcome: A scope statement with explicit inclusions, exclusions and boundary-test questions.

Inputs: [PROJECT] charter/SOW, [SOURCE_DATA] requirements docs, [OBJECTIVE]

PROMPT

```
CONTEXT: Defining the scope baseline for [PROJECT] under [OBJECTIVE].
OBJECTIVE: Produce a scope statement with explicit in-scope and out-of-scope items and a reusable boundary test.
INPUTS: Charter/SOW, requirements documentation, prior scope discussions if recorded in [SOURCE_DATA].
CONSTRAINTS: Every in-scope item must trace to the charter/SOW or an approved change. Do not include items only implied by conversation.
DECISION RULES: An item is IN-SCOPE only if it appears in [SOURCE_DATA]; otherwise it is OUT-OF-SCOPE or DATA GAP (needs sponsor clarification).
OUTPUT CONTRACT: Sections — In Scope (bulleted, each with source reference) | Out of Scope (explicit) | Boundary Test (a yes/no question to classify future requests) | Open Scope Questions.
VALIDATION: Each in-scope bullet cites its source document/clause.
ESCALATION: Items neither clearly in nor out route to sponsor for explicit ruling before baseline sign-off.
```

Output: Scope statement with cited in/out items and a reusable boundary test.

Validate: Every in-scope item cites a source document reference.

Next: Baseline in the schedule; use boundary test in PM-06 Change Impact Analysis.

LEVEL L1 HUMAN GATE Yes

PM-02 Schedule Health Check

Use when: You need an objective read on whether the schedule is realistic and on track.

Outcome: A diagnostic of schedule health: float erosion, logic issues, and realistic forecast.

Inputs: Current schedule/Gantt data, baseline schedule, [DATE]

PROMPT

```
CONTEXT: Reviewing schedule health for [PROJECT] against baseline as of [DATE].
OBJECTIVE: Diagnose schedule realism — float erosion, missing dependencies, unrealistic durations — and produce a defensible forecast.
INPUTS: Current schedule vs baseline, resource assignments, actual progress to date.
CONSTRAINTS: Base forecast on documented progress rates, not optimistic assumptions of catch-up.
DECISION RULES: Flag task as AT RISK if float has eroded >30% since baseline. Flag as SUSPECT if duration was never re-validated against actual team velocity.
OUTPUT CONTRACT: Table [Task/Milestone | Baseline Date | Current Forecast | Variance | Float Status | Flag] + one-paragraph forecast confidence statement.
VALIDATION: Forecast dates recalculated from actual progress rate, not carried forward from baseline unchanged.
ESCALATION: If baseline data is missing or unreconciled, flag DATA GAP before issuing a forecast.
```

Output: Schedule diagnostic table with variance, float status, and a forecast confidence statement.

Validate: Forecast derived from actual progress rate, not baseline optimism.

Next: Feed into PM-10 Status Report Generator or PM-11 Recovery Plan.

LEVEL L1 HUMAN GATE No

PM-03 Cost / Budget Variance Analysis

Use when: You need to explain budget variance with evidence, not just report the number.

Outcome: A variance analysis distinguishing timing variance from true cost overrun.

Inputs: Budget baseline, actuals/commitments to date, [DATE], [THRESHOLD] for variance escalation

PROMPT

```
CONTEXT: Analyzing cost performance for [PROJECT] as of [DATE] against baseline budget.
OBJECTIVE: Explain budget variance and distinguish timing variance (spend shifted, not lost) from
true overrun.
INPUTS: Budget baseline by category/phase, actuals and committed spend to date.
CONSTRAINTS: Do not attribute variance to a cause without a specific line-item or change record
supporting it.
DECISION RULES: Classify variance as TIMING if offsetting underspend exists elsewhere; as OVERRUN
if no offset and trend is worsening. Escalate if projected overrun exceeds [THRESHOLD].
OUTPUT CONTRACT: Table [Category | Baseline | Actual+Committed | Variance | Classification |
Cause] + Estimate at Completion (EAC) with method stated.
VALIDATION: EAC calculation method (e.g., CPI-based, bottom-up) explicitly stated and
reproducible.
ESCALATION: Overrun beyond [THRESHOLD] routes to PM-11 Recovery Plan and sponsor notification.
```

Output: Cost variance table with timing-vs-overrun classification and a stated-method EAC.

Validate: EAC method is explicit and independently reproducible from the data.

Next: Overrun triggers PM-11 Recovery Plan; otherwise feeds PM-10 Status Report.

LEVEL L1 HUMAN GATE Conditional

PM-04 RAID Log Build

Use when: Starting or re-baselining a project and you need a structured, first-pass RAID log.

Outcome: An initial RAID log built from source documents, not blank templates.

Inputs: [SOURCE_DATA] charter, discovery notes, stakeholder input, contracts

PROMPT

```
CONTEXT: Building the initial RAID log for [PROJECT] from available discovery material.
OBJECTIVE: Extract candidate Risks, Assumptions, Issues and Dependencies directly from source
material.
INPUTS: Charter, discovery/workshop notes, contract terms, any prior correspondence in
[SOURCE_DATA].
CONSTRAINTS: Every entry must trace to a specific source passage. Do not generate
generic/boilerplate risks not grounded in the material.
DECISION RULES: Classify each finding strictly: Risk (uncertain, future, negative), Assumption
(unverified belief treated as true), Issue (already occurred), Dependency (external reliance).
OUTPUT CONTRACT: Table [ID | Type | Statement | Source Reference | Suggested Owner | Suggested
Priority].
VALIDATION: Spot-check classification logic on 3 entries — confirm none is mislabeled (e.g., an
issue mislabeled as a risk).
ESCALATION: Items with no clear source basis are dropped, not included as filler.
```

Output: Source-grounded initial RAID log ready for owner assignment.

Validate: Sampled entries confirmed correctly classified by type.

Next: Assign owners; feed high-priority risks into PM-05 Risk Response Planning.

LEVEL L1 HUMAN GATE No

PM-05 Risk Response Planning

Use when: Identified risks need formal response strategies, not just a log entry.

Outcome: A response plan per priority risk using the standard avoid/mitigate/transfer/accept framework.

Inputs: Prioritized risk list, [RISK_TOLERANCE], [CONSTRAINTS] (budget/schedule)

PROMPT

```
CONTEXT: Defining risk responses for the top risks on [PROJECT] against [RISK_TOLERANCE].
OBJECTIVE: Assign a response strategy and concrete action per priority risk.
INPUTS: Prioritized risk list (from PM-04 or scoring), budget/schedule constraints.
CONSTRAINTS: Response strategy must be one of Avoid/Mitigate/Transfer/Accept — justify the
choice, don't default to Mitigate.
DECISION RULES: Choose Avoid if root cause is eliminable within [CONSTRAINTS]; Transfer if
insurable/contractable; Accept only if within [RISK_TOLERANCE] and cost-to-mitigate exceeds
impact; else Mitigate.
OUTPUT CONTRACT: Table [Risk | Strategy | Justification | Specific Action | Owner | Trigger for
Contingency Plan | Residual Risk Rating].
VALIDATION: Each strategy choice justified against the decision rule, not asserted without
reasoning.
ESCALATION: Any risk where chosen strategy is Accept but score exceeds [RISK_TOLERANCE] must be
escalated for sponsor sign-off.
```

Output: Risk response plan with strategy, action, owner and residual rating per risk.

Validate: Strategy choice justified per decision rule for each risk.

Next: Track actions in RAID; residual risks feed next PM-02/PM-03 review cycle.

LEVEL L2 HUMAN GATE Conditional

PM-06 Change Request Impact Analysis

Use when: A change request has landed and you need a rigorous impact analysis before approval.

Outcome: A structured impact analysis across scope, schedule, cost and risk with a clear recommendation.

Inputs: Change request description, scope baseline (PM-01), schedule and budget baselines

PROMPT

```
CONTEXT: Assessing a change request on [PROJECT]: <describe change>.
OBJECTIVE: Quantify impact on scope, schedule, cost and risk, and recommend approve/reject/defer.
INPUTS: Change description, current scope/schedule/budget baselines.
CONSTRAINTS: Quantify schedule and cost impact numerically wherever baseline data allows; do not
use vague terms like "some delay."
DECISION RULES: Recommend APPROVE if impact is within [CONSTRAINTS] and business value is stated;
DEFER if value is unclear; REJECT if impact breaches contractual or budget limits without funded
offset.
OUTPUT CONTRACT: Sections — Change Summary | Scope Impact | Schedule Impact (days) | Cost Impact
(currency) | Risk Impact | Recommendation with rationale.
VALIDATION: Confirm schedule/cost figures are calculated against the current baseline, not
estimated freehand.
ESCALATION: Changes exceeding governance approval thresholds route to steering/sponsor — human
decision required.
```

Output: Quantified change impact analysis with an approve/reject/defer recommendation.

Validate: Cost and schedule impact figures calculated against current baseline data.

Next: Approved changes update PM-01 scope baseline and PM-02 schedule.

LEVEL L1 HUMAN GATE Yes

PM-07 Resource Plan & Allocation Check

Use when: You need to verify the resource plan is realistic against actual demand and availability.

Outcome: A validated resource plan flagging over-allocation and unfilled roles.

Inputs: Resource plan, role/skill demand by phase, known availability constraints

PROMPT

```
CONTEXT: Validating the resource plan for [PROJECT] against forecast demand by phase.
OBJECTIVE: Identify over-allocation, unfilled roles, and single points of failure (one person
covering a critical role with no backup).
INPUTS: Resource plan/roster, demand by phase, known leave or competing-commitment data.
CONSTRAINTS: Flag over-allocation only where documented commitment exceeds 100% FTE in a given
period; do not assume unstated part-time availability.
DECISION RULES: Flag SPOF if a critical-path task's sole assigned resource has no named backup.
Flag OVER-ALLOCATED if total commitment > 100% in any period.
OUTPUT CONTRACT: Table [Resource | Role | Period | Allocation % | Flag | Recommended Action].
VALIDATION: Allocation percentages sum correctly per person per period before flagging.
ESCALATION: SPOF on critical-path roles routes to PM sponsor for backfill/contingency decision.
```

Output: Resource allocation table flagging over-allocation and single points of failure.

Validate: Allocation totals independently summed and checked before flags applied.

Next: Over-allocation feeds PM-11 Recovery Plan if schedule impact is confirmed.

LEVEL L1 HUMAN GATE No

PM-08 Stakeholder Analysis & Engagement Plan

Use when: You need a structured stakeholder map to plan communication and manage influence.

Outcome: A power/interest-mapped stakeholder register with tailored engagement actions.

Inputs: Known stakeholders, [SOURCE_DATA] org context, [OBJECTIVE]

PROMPT

```
CONTEXT: Building the stakeholder engagement plan for [PROJECT] with objective [OBJECTIVE].
OBJECTIVE: Map stakeholders by power/interest and define a specific engagement action per
stakeholder.
INPUTS: Stakeholder list, known org roles/context, current sentiment if known.
CONSTRAINTS: Do not assume sentiment (Supportive/Neutral/Resistant) without evidence; mark
unknown sentiment as DATA GAP.
DECISION RULES: High power + high interest → Manage Closely; high power + low interest → Keep
Satisfied; low power + high interest → Keep Informed; low power + low interest → Monitor.
OUTPUT CONTRACT: Table [Stakeholder | Power | Interest | Quadrant | Current Sentiment |
Engagement Action | Frequency].
VALIDATION: Confirm quadrant assignment follows the decision rule exactly for each stakeholder.
ESCALATION: Stakeholders marked Resistant with High Power route to PM sponsor for a joint
engagement approach.
```

Output: Power/interest stakeholder map with tailored engagement actions.

Validate: Quadrant placement checked against the stated power/interest decision rule.

Next: High-power resistant stakeholders feed a dedicated engagement conversation before go-live.

LEVEL L1 HUMAN GATE Conditional

PM-09 Governance Cadence Design

Use when: Standing up a new project or fixing a governance structure that isn't working.

Outcome: A right-sized governance cadence matched to project risk and complexity, not a generic template.

Inputs: [PROJECT] size/risk profile, [AUDIENCE] (governance bodies), current pain points

PROMPT

```
CONTEXT: Designing/re-designing the governance cadence for [PROJECT], audience [AUDIENCE].
OBJECTIVE: Define forums, frequency, membership and decision rights sized to actual project risk
and complexity.
INPUTS: Project size/complexity/risk profile, current governance pain points if any.
CONSTRAINTS: Do not default to weekly steering for a low-risk project, or monthly for a high-risk
one — size cadence to [RISK_TOLERANCE] and complexity.
DECISION RULES: High risk/complexity → weekly operational + fortnightly steering. Medium →
fortnightly operational + monthly steering. Low → monthly combined forum.
OUTPUT CONTRACT: Table [Forum | Purpose | Frequency | Membership | Decision Rights | Inputs
Required] + RACI summary for key governance decisions.
VALIDATION: Confirm cadence choice matches the stated risk/complexity tier, not a default
template.
ESCALATION: Disputed decision rights (unclear who can approve what) route to sponsor for explicit
ruling before launch.
```

Output: Right-sized governance cadence with forums, membership and decision rights.

Validate: Cadence tier matches the project's documented risk/complexity level.

Next: Publish as the governance charter annex; revisit at each major phase gate.

LEVEL L1 HUMAN GATE Yes

PM-10 Status Report Generator

Use when: Regular reporting cycle; you need a consistent, decision-oriented status report fast.

Outcome: A structured status report leading with exceptions and decisions, not a chronological recap.

Inputs: Latest schedule/cost/risk data, [AUDIENCE], [DATE]

PROMPT

```
CONTEXT: Producing the [DATE] status report for [PROJECT], audience [AUDIENCE].
OBJECTIVE: Report status in a way that leads with exceptions and required decisions.
INPUTS: Current schedule variance, cost variance, top risks/issues, milestone status.
CONSTRAINTS: Report only changes and exceptions in the narrative; full detail belongs in appendix
tables, not prose.
DECISION RULES: Overall RAG = RED if any committed milestone has slipped without an approved
recovery plan; otherwise apply TDM-01-style dimension logic.
OUTPUT CONTRACT: Sections — Overall RAG + 1-sentence why | Key Decisions Needed | Milestone
Status Table | Top Risks/Issues | Budget Snapshot | Next Period Focus.
VALIDATION: Confirm the RAG rating's stated "why" matches the milestone table data.
ESCALATION: Any RED rating without a listed recovery plan link triggers a request for PM-11.
```

Output: Decision-oriented status report ready for [AUDIENCE] distribution.

Validate: RAG rationale cross-checked against the milestone status table.

Next: RED items without a plan trigger PM-11 Recovery Plan.

LEVEL L1 HUMAN GATE No

PM-11 Recovery Plan

Use when: Project is off baseline on schedule, cost or scope and needs a credible path back.

Outcome: A recovery plan with prioritized actions and a revised, defensible baseline.

Inputs: Variance analysis (PM-02/PM-03), root cause if known, [CONSTRAINTS]

PROMPT

```
CONTEXT: [PROJECT] is off baseline on <dimension(s)>. Root cause: <summary>. Constraints:
[CONSTRAINTS].
OBJECTIVE: Define recovery actions and produce a revised, credible baseline.
INPUTS: Variance analysis, root cause summary, hard constraints (budget cap, contractual dates).
CONSTRAINTS: Every action must target the stated root cause. Do not propose "add more resources"
without confirming budget/[CONSTRAINTS] allows it.
DECISION RULES: Sequence actions by impact-to-effort ratio; only include actions achievable
within [CONSTRAINTS].
OUTPUT CONTRACT: Table [Action | Root Cause Addressed | Owner | Target Date | Expected Recovery]
+ Revised Baseline (date/cost/scope) + Residual Risk statement.
VALIDATION: Revised baseline figures recalculated from the recovery actions, not asserted.
ESCALATION: If full recovery isn't achievable within [CONSTRAINTS], state the gap explicitly and
escalate the trade-off decision to sponsor.
```

Output: Recovery action plan with a recalculated, defensible revised baseline.

Validate: Revised baseline figures are recalculated from stated actions, not asserted.

Next: Get sponsor sign-off; re-baseline PM-02/PM-03 tracking against the new plan.

LEVEL L2 HUMAN GATE Yes

PM-12 Benefits & Lessons Learned Review

Use when: At a phase gate or project close, to capture what value was delivered and what to do differently.

Outcome: A benefits realization check against the business case, plus actionable lessons.

Inputs: Original business case/benefits case, actuals to date, project retrospective input

PROMPT

```
CONTEXT: Reviewing benefits realization and lessons learned for [PROJECT] at <phase gate/close>.
OBJECTIVE: Compare realized benefits against the original business case and extract specific,
reusable lessons.
INPUTS: Original business case with target benefits, actual measured outcomes, delivery team
retrospective input.
CONSTRAINTS: Only claim a benefit is realized where a measured outcome exists; do not credit
benefits still "expected."
DECISION RULES: Mark benefit REALIZED if measured actual meets or exceeds target; PARTIAL if
measured but below target; NOT YET MEASURABLE if too early.
OUTPUT CONTRACT: Table [Benefit | Target | Actual | Status | Evidence Source] + Lessons Learned
table [Lesson | Category (Process/Tech/People/Governance) | Recommended Change | Owner for
Follow-up].
VALIDATION: Every REALIZED status backed by a cited measurement source, not stakeholder
impression.
ESCALATION: PARTIAL/unrealized benefits route to sponsor and PMO for portfolio-level tracking.
```

Output: Benefits realization scorecard plus a structured, actionable lessons-learned log.

Validate: Realized status is backed by a cited measurement, not opinion.

Next: Route unrealized benefits to PMO-06 Benefits Realization Framework for portfolio tracking.

LEVEL L2 HUMAN GATE Conditional

Program Manager / TPM (TPM)

Cross-project orchestration: dependencies, integrated roadmaps, program risk and executive trade-offs.

TPM-01 Cross-Project Dependency Map

Use when: Multiple projects share dependencies and you need a single map to manage sequencing risk.

Outcome: A consolidated cross-project dependency map with directionality and risk flags.

Inputs: [PROGRAM] constituent project plans, known integration/handoff points

PROMPT

```
CONTEXT: Mapping cross-project dependencies within [PROGRAM].
OBJECTIVE: Consolidate dependencies across constituent projects into one map with direction and
timing.
INPUTS: Individual project schedules/dependency registers, known shared systems or teams.
CONSTRAINTS: Include only documented dependencies; do not infer a dependency purely from shared
vendor or system name without confirmation.
DECISION RULES: Flag HIGH RISK if a dependency has no buffer between provider completion and
consumer start date.
OUTPUT CONTRACT: Table [Providing Project | Consuming Project | Dependency | Required By | Buffer
(days) | Risk Flag] + narrative on the tightest 3 dependencies.
VALIDATION: Buffer calculated as (consumer start date – provider committed finish date); recheck
arithmetic.
ESCALATION: Zero/negative buffer dependencies escalate to TPM-11 Escalation and program risk log.
```

Output: Cross-project dependency map with calculated buffers and risk flags.

Validate: Buffer days independently recalculated from the two linked dates.

Next: High-risk dependencies feed TPM-04 Critical Path and TPM-03 Program Risk.

LEVEL L2 HUMAN GATE No

TPM-02 Integrated Roadmap Builder

Use when: You need one roadmap that reconciles multiple project timelines into a coherent program view.

Outcome: A single integrated roadmap that reconciles conflicting or overlapping project timelines.

Inputs: Individual project roadmaps, [BUSINESS_OUTCOME], major milestones

PROMPT

```
CONTEXT: Building the integrated roadmap for [PROGRAM] to deliver [BUSINESS_OUTCOME].
OBJECTIVE: Reconcile individual project roadmaps into one coherent, sequenced program roadmap.
INPUTS: Project-level roadmaps/milestones, known dependency map (TPM-01).
CONSTRAINTS: Where two projects claim conflicting dates for a shared milestone, flag the conflict
rather than silently picking one.
DECISION RULES: Sequence by dependency (TPM-01) first, then by business value priority where no
dependency exists.
OUTPUT CONTRACT: Timeline table [Milestone | Owning Project | Quarter/Date | Dependencies |
Status] + a list of unresolved date conflicts requiring reconciliation.
VALIDATION: Every milestone sequencing decision traces to either a dependency constraint or a
stated priority rule.
ESCALATION: Unresolved date conflicts route to program governance for reconciliation before
roadmap publication.
```

Output: Integrated program roadmap with an explicit conflict list.

Validate: Sequencing decisions traced to dependency or priority rule, not default ordering.

Next: Publish via TPM-06 Executive Reporting; unresolved conflicts to TPM-11 Escalation.

LEVEL L2 HUMAN GATE Yes

TPM-03 Program Risk Aggregation

Use when: Individual project risks need to be rolled up to see program-level exposure.

Outcome: An aggregated program risk view that surfaces systemic and compounding risks, not just a merged list.

Inputs: Constituent project risk registers, [RISK_TOLERANCE] at program level

PROMPT

```
CONTEXT: Aggregating risk across constituent projects in [PROGRAM] against program-level
[RISK_TOLERANCE].
OBJECTIVE: Identify systemic risks (affecting multiple projects) and compounding risk
combinations, not just merge lists.
INPUTS: Project-level risk registers with scores.
CONSTRAINTS: A risk is SYSTEMIC only if it appears independently in 2+ project registers or
shares a common root cause across projects.
DECISION RULES: Elevate to program risk register if SYSTEMIC, or if a single project risk's
impact would breach program-level [RISK_TOLERANCE] if realized.
OUTPUT CONTRACT: Table [Risk | Type (Systemic/Project-specific escalated) | Affected Projects |
Aggregate Score | Program Response Owner].
VALIDATION: Systemic classification checked against the stated 2+ project rule.
ESCALATION: Systemic risks above [RISK_TOLERANCE] route to TPM-06 Executive Reporting this cycle.
```

Output: Program-level risk register distinguishing systemic from escalated project risks.

Validate: Systemic tagging verified against the multi-project evidence rule.

Next: Feed top items into TPM-06 Executive Reporting and TPM-11 Escalation.

LEVEL L2 HUMAN GATE No

TPM-04 Program Critical Path Analysis

Use when: You need to know which single chain of work, across projects, determines the program end date.

Outcome: An identified cross-project critical chain with the true program-level constraint.

Inputs: Integrated roadmap (TPM-02), dependency map (TPM-01), float/slack data per project

PROMPT

```
CONTEXT: Determining the program-level critical path for [PROGRAM].
OBJECTIVE: Identify the single chain of cross-project tasks/dependencies that determines the
program completion date.
INPUTS: Integrated roadmap, dependency map, task-level float where available.
CONSTRAINTS: Critical path must be traced task-by-task across project boundaries, not assumed at
the milestone level only.
DECISION RULES: The critical chain is the path with cumulative float = 0 (or lowest) end-to-end
across all constituent projects.
OUTPUT CONTRACT: Ordered list of the critical chain [Task | Owning Project | Float | Constraint
Type] + statement of the program end-date driver.
VALIDATION: Confirm the identified chain has the lowest cumulative float versus at least one
alternative chain checked.
ESCALATION: If float data is unavailable for any project, flag DATA GAP for that segment rather
than assuming zero float.
```

Output: Cross-project critical chain with the named program end-date driver.

Validate: Chain float compared against at least one alternative path to confirm it's lowest.

Next: Monitor chain tasks weekly; any slip triggers TPM-10 Scenario Planning.

LEVEL L2 HUMAN GATE No

TPM-05 Program Capacity Analysis

Use when: Shared resource pools across projects create contention you need to quantify.

Outcome: A program-level capacity view showing contention across shared pools, by period.

Inputs: Shared resource pool data, per-project demand forecasts, [THRESHOLD] utilization

PROMPT

```
CONTEXT: Analyzing shared-resource capacity contention across projects in [PROGRAM].
OBJECTIVE: Quantify capacity contention on shared pools (e.g., architecture, security, shared
engineering) by period.
INPUTS: Shared pool headcount/capacity, per-project demand forecasts referencing that pool.
CONSTRAINTS: Aggregate demand only from documented project forecasts; do not assume unstated
demand.
DECISION RULES: Flag CONTENTION if aggregate demand > [THRESHOLD] of pool capacity in any period.
OUTPUT CONTRACT: Table [Shared Pool | Period | Aggregate Demand | Capacity | Utilization % |
Contending Projects | Flag].
VALIDATION: Aggregate demand independently summed across the listed contending projects.
ESCALATION: CONTENTION periods route to TPM-08 Trade-off Analysis for prioritization decision.
```

Output: Shared-pool capacity contention table by period with contending projects named.

Validate: Demand totals independently summed and checked before flagging.

Next: Contention periods route to TPM-08 Trade-off Analysis.

LEVEL L2 HUMAN GATE No

TPM-06 Executive Program Report

Use when: Regular program-level reporting to sponsors/executive steering.

Outcome: A single executive report synthesizing program status, decisions and trajectory.

Inputs: Aggregated status from constituent projects, program risk (TPM-03), [AUDIENCE], [DATE]

PROMPT

```
CONTEXT: Producing the [DATE] executive report for [PROGRAM], audience [AUDIENCE].
OBJECTIVE: Synthesize constituent project status into a program-level narrative with clear
decisions needed.
INPUTS: Consolidated project statuses, program risk register, integrated roadmap status.
CONSTRAINTS: Report at the program level; drill-down project detail belongs in an appendix, not
the executive narrative.
DECISION RULES: Program RAG = RED if the critical chain (TPM-04) has slipped without an approved
recovery plan.
OUTPUT CONTRACT: Sections — Program RAG + why | Decisions Needed | Progress Against
[BUSINESS_OUTCOME] | Top Program Risks | Budget Snapshot | Appendix (project detail table).
VALIDATION: Program RAG rationale checked against critical-chain status specifically, not an
average of project statuses.
ESCALATION: Any RED without a linked recovery plan triggers a request to the responsible TDM/PM.
```

Output: Executive program report anchored to business outcomes and decisions.

Validate: RAG rationale tied to critical-chain status, not a simple average.

Next: Distribute pre-read before steering; log decisions in program RAID.

LEVEL L1 HUMAN GATE Yes

TPM-07 Program Governance Design

Use when: Standing up governance across multiple projects reporting into one program.

Outcome: A governance structure with clear program vs project decision rights, avoiding duplication.

Inputs: Number/type of constituent projects, [AUDIENCE], existing project-level governance

PROMPT

```
CONTEXT: Designing program governance for [PROGRAM] spanning multiple constituent projects.
OBJECTIVE: Define program-level forums and decision rights that complement (not duplicate)
project-level governance.
INPUTS: List of constituent projects and their existing governance, program risk profile.
CONSTRAINTS: Program governance should only take decisions that are cross-project in nature;
project-specific decisions stay at project level.
DECISION RULES: A decision belongs at program level if it affects 2+ constituent projects' scope,
schedule or budget.
OUTPUT CONTRACT: RACI-style table [Decision Type | Program Level or Project Level | Forum |
Frequency | Approval Authority].
VALIDATION: Spot-check 3 decision types to confirm the level assignment follows the 2+ project
rule.
ESCALATION: Ambiguous decision-rights cases route to program sponsor for an explicit ruling in
the governance charter.
```

Output: Program governance model with explicit program-vs-project decision rights.

Validate: Decision-level assignment checked against the cross-project impact rule.

Next: Publish as program governance charter annex.

LEVEL L1 HUMAN GATE Yes

TPM-08 Trade-off Analysis

Use when: Competing priorities across projects require an explicit, defensible trade-off decision.

Outcome: A structured trade-off analysis with quantified consequences per option, not opinion.

Inputs: The competing priorities/asks, [BUSINESS_OUTCOME], [CONSTRAINTS]

PROMPT

```
CONTEXT: [PROGRAM] faces a trade-off: <describe competing priorities/asks> against
[BUSINESS_OUTCOME] and [CONSTRAINTS].
OBJECTIVE: Quantify the consequence of each trade-off option to support a defensible decision.
INPUTS: Description of competing asks, relevant capacity/dependency/budget data.
CONSTRAINTS: Every option must show what is gained and what is given up — no option framed as
"free."
DECISION RULES: Rank options by contribution to [BUSINESS_OUTCOME] per unit of [CONSTRAINTS]
consumed (e.g., value per resource-week).
OUTPUT CONTRACT: Decision matrix [Option | Gain | Cost/What's Given Up | Risk | Contribution to
[BUSINESS_OUTCOME]] + recommendation with rationale.
VALIDATION: Confirm every option lists an explicit cost, not just a benefit.
ESCALATION: Trade-offs affecting committed customer dates or contracts require sponsor/steering
sign-off.
```

Output: Quantified trade-off decision matrix with a clear recommendation.

Validate: Every option lists both gain and explicit cost — no free options.

Next: Escalate to TPM-06 Executive Report if trade-off affects committed dates.

LEVEL L1 HUMAN GATE Yes

TPM-09 Benefits Tracking (Program Level)

Use when: You need to track whether the program is on pace to deliver its intended business outcomes.

Outcome: A benefits tracking view tied to [BUSINESS_OUTCOME], separate from project completion status.

Inputs: Program business case, measured outcomes to date, [BUSINESS_OUTCOME] targets

PROMPT

```
CONTEXT: Tracking benefits realization for [PROGRAM] against [BUSINESS_OUTCOME].
OBJECTIVE: Report benefits progress independent of project completion status — projects can
finish without benefits landing.
INPUTS: Program business case with target benefits, measured actuals, project delivery status.
CONSTRAINTS: Do not conflate "project delivered" with "benefit realized" — they must be tracked
as separate columns.
DECISION RULES: Mark ON TRACK if measured trajectory meets the interim target for this point in
the timeline; else AT RISK.
OUTPUT CONTRACT: Table [Benefit | Target | Interim Milestone Target | Actual to Date | Status |
Enabling Project(s) Delivered? (Y/N)].
VALIDATION: Confirm no benefit is marked ON TRACK using project-completion status as a proxy for
actual measured benefit.
ESCALATION: AT RISK benefits with all enabling projects delivered indicate an adoption/benefits-
realization gap — escalate to sponsor, not delivery team.
```

Output: Benefits tracking table decoupled from project delivery status.

Validate: No benefit status inferred from project completion alone.

Next: Feed into PMO-06 Benefits Realization Framework for portfolio rollup.

LEVEL L2 HUMAN GATE No

TPM-10 Scenario Planning

Use when: A major risk or decision point requires comparing multiple future paths before committing.

Outcome: A structured comparison of 2-3 credible scenarios with trigger conditions and recommended path.

Inputs: The decision point/uncertainty, plausible scenarios, [SUCCESS_CRITERIA]

PROMPT

```
CONTEXT: [PROGRAM] faces a key uncertainty: <describe>. Evaluating scenarios against
[SUCCESS_CRITERIA].
OBJECTIVE: Compare 2–3 plausible scenarios with explicit trigger conditions and consequences for
each.
INPUTS: Description of the uncertainty, known constraints, [SUCCESS_CRITERIA] for the program.
CONSTRAINTS: Each scenario must include a specific, observable trigger condition — not a vague
"if things go badly."
DECISION RULES: Recommend the scenario response that best satisfies [SUCCESS_CRITERIA] under the
most probable trigger, while noting the fallback if a less probable trigger fires.
OUTPUT CONTRACT: Table [Scenario | Trigger Condition | Probability (High/Med/Low, with basis) |
Program Impact | Recommended Response] + recommended monitoring signal.
VALIDATION: Confirm each trigger condition is observable/measurable, not subjective.
ESCALATION: If probability cannot be reasonably estimated from evidence, mark DATA GAP rather
than guessing a number.
```

Output: Scenario comparison table with observable triggers and recommended responses.

Validate: Each trigger condition confirmed observable and specific, not vague.

Next: Assign a monitoring owner for each trigger signal.

LEVEL L2 HUMAN GATE Yes

TPM-11 Program Escalation Brief

Use when: A cross-project issue exceeds program-level resolution authority and needs sponsor decision.

Outcome: A one-page program escalation brief enabling a fast cross-project decision.

Inputs: The issue, affected projects, options considered, [DATE] needed by

PROMPT

```
CONTEXT: Escalating a program-level issue on [PROGRAM] affecting multiple constituent projects,
decision needed by [DATE].
OBJECTIVE: Produce a one-page brief enabling a fast, informed cross-project decision.
INPUTS: Issue description, affected projects and their individual positions, options considered.
CONSTRAINTS: Present the impact on [BUSINESS_OUTCOME], not just individual project impact.
DECISION RULES: Include a "cost of no decision by [DATE]" line at the program level, aggregating
affected-project impacts.
OUTPUT CONTRACT: Sections — Situation | Affected Projects & Impact | Options (min. 2, with
program-level trade-offs) | Recommendation | Decision Owner | Needed By [DATE].
VALIDATION: Confirm the brief aggregates impact across all named affected projects, not just the
originating one.
ESCALATION: This is the escalation artifact itself — route to program sponsor; human decision
required.
```

Output: One-page program-level escalation brief with cross-project impact.

Validate: Impact aggregated across every affected project named in the issue.

Next: Route to sponsor; log decision in program risk/RAID once made.

LEVEL L1 HUMAN GATE Yes

PMO / Transformation Lead (PMO)

Portfolio-level governance, KPI discipline and transformation oversight grounded in aggregated evidence.

PMO-01 Portfolio Health Assessment

Use when: You need a portfolio-wide health view rolled up from individual project/program statuses.

Outcome: A portfolio health view that avoids RAG-washing — surfaces true worst-case exposure, not averages.

Inputs: Individual project/program RAG statuses, [THRESHOLD] for portfolio-level escalation

PROMPT

```
CONTEXT: Assessing portfolio health across [PROGRAM]s/projects under the PMO.
OBJECTIVE: Roll up individual statuses into a portfolio view that preserves visibility of worst-case exposure, not an averaged/smoothed picture.
INPUTS: Individual project/program RAG statuses with underlying evidence, budget/schedule variance data.
CONSTRAINTS: Do not average RAG ratings into a portfolio "mostly green" statement if any project poses [THRESHOLD]-exceeding exposure — surface it explicitly.
DECISION RULES: Portfolio status = RED if any single project's exposure (budget, schedule, or strategic) exceeds [THRESHOLD], regardless of the count of green projects.
OUTPUT CONTRACT: Portfolio summary (count by RAG) + explicit "Highest Exposure Items" list (not hidden by aggregate averaging) + trend vs last period.
VALIDATION: Confirm the portfolio-level status reflects the worst-case exposure rule, not a simple majority/average.
ESCALATION: RED portfolio status routes to executive sponsor briefing this cycle, not the next scheduled review.
```

Output: Exposure-preserving portfolio health view with explicit highest-risk items.

Validate: Portfolio status confirmed to reflect worst-case exposure rule, not averaging.

Next: Feed into PMO-08 Executive Portfolio Report.

LEVEL L2 HUMAN GATE No

PMO-02 Governance Framework Design

Use when: Designing or refreshing portfolio-level governance across multiple programs/projects.

Outcome: A governance framework with clear decision rights and escalation thresholds, sized to portfolio risk.

Inputs: Portfolio composition/risk profile, [AUDIENCE], current governance pain points

PROMPT

```
CONTEXT: Designing portfolio governance framework across [PROGRAM]s for [CUSTOMER]/the organization.
OBJECTIVE: Define governance forums, decision rights and escalation thresholds sized to actual portfolio risk and complexity, avoiding one-size-fits-all governance.
INPUTS: Portfolio composition (number/size/risk of constituent programs), current pain points (e.g., decision bottlenecks, unclear authority).
CONSTRAINTS: Escalation thresholds must be quantified (e.g., budget variance %, schedule slip days) — not left to subjective judgment of "significant."
DECISION RULES: High-risk/high-value programs get tailored, more frequent governance; low-risk/small programs use a lightweight standard template — do not apply uniform heavy governance across all.
OUTPUT CONTRACT: Table [Program Tier | Governance Forum | Frequency | Escalation Threshold | Decision Rights] + tiering criteria used.
VALIDATION: Confirm tiering criteria are applied consistently and quantitatively across listed programs.
ESCALATION: Disputed program tiering routes to PMO leadership for a final ruling.
```

Output: Risk-tiered portfolio governance framework with quantified escalation thresholds.

Validate: Tiering criteria confirmed applied consistently and quantitatively.

Next: Publish as the PMO governance charter; review annually.

LEVEL L1 HUMAN GATE Yes

PMO-03 KPI / Metrics Framework

Use when: Defining what the PMO should actually measure across the portfolio to drive decisions, not just report.

Outcome: A KPI framework where every metric ties to a specific decision it informs, avoiding vanity metrics.

Inputs: [BUSINESS_OUTCOME] priorities, current reporting pain points, available data sources

PROMPT

```
CONTEXT: Defining the PMO KPI/metrics framework across [PROGRAM]s in service of
[BUSINESS_OUTCOME].
OBJECTIVE: Define a compact metric set where every metric is tied to a specific decision or
action it informs.
INPUTS: Business outcome priorities, current reporting complaints (e.g., "too many metrics, no
clear action"), available data sources.
CONSTRAINTS: Reject any metric that cannot be tied to a specific "if this metric crosses X, we do
Y" decision rule — this excludes vanity/context-only metrics from the core set.
DECISION RULES: Include a metric in the CORE set only if paired with a decision rule; metrics
without one go to a CONTEXTUAL (informational only) set.
OUTPUT CONTRACT: Table [Metric | Data Source | Core or Contextual | Decision Rule (if Core) |
Owner].
VALIDATION: Confirm every Core-classified metric has an explicit, checkable decision rule.
ESCALATION: Metrics with no available data source route to an instrumentation request before
adoption.
```

Output: Decision-tied KPI framework distinguishing core metrics from contextual ones.

Validate: Every core metric confirmed to have an explicit, checkable decision rule.

Next: Instrument missing sources; review framework at each portfolio cycle.

LEVEL L1 HUMAN GATE No

PMO-04 PMO Maturity Assessment

Use when: Assessing the PMO's own operating maturity to identify improvement priorities.

Outcome: A maturity assessment against a recognized model, evidence-based, with a prioritized improvement path.

Inputs: PMO current practices/processes, [SOURCE_DATA] on outcomes achieved, chosen maturity model reference

PROMPT

```
CONTEXT: Assessing PMO maturity for [CUSTOMER]/the organization against a recognized maturity
model.
OBJECTIVE: Rate current maturity per dimension with evidence, and prioritize improvement actions
by impact on delivery outcomes.
INPUTS: Current PMO processes/practices per dimension, evidence of outcomes (delivery
predictability, benefits realization track record).
CONSTRAINTS: Rate maturity from evidence of actual practice and outcome, not from documented
process existence alone (a documented-but-unused process is not "mature").
DECISION RULES: Prioritize improvement actions targeting dimensions with both LOW maturity and
demonstrated linkage to poor delivery outcomes (e.g., low risk-management maturity correlating
with frequent escalations).
OUTPUT CONTRACT: Table [Dimension | Maturity Level | Evidence | Linked Outcome Impact | Priority]
+ prioritized improvement roadmap.
VALIDATION: Maturity ratings checked against actual practice/outcome evidence, not documentation
existence alone.
ESCALATION: Low-maturity dimensions with executive visibility (e.g., benefits realization) route
to sponsor for an improvement investment discussion.
```

Output: Evidence-based PMO maturity assessment with an outcome-linked improvement roadmap.

Validate: Maturity ratings checked against actual practice/outcome evidence, not documentation.

Next: Feed into PMO-05 Transformation Roadmap for the PMO's own improvement.

LEVEL L2 HUMAN GATE No

PMO-05 Transformation Roadmap

Use when: Sequencing organizational/PMO transformation initiatives into an executable roadmap.

Outcome: A phased transformation roadmap sequenced by readiness and dependency, with named milestones.

Inputs: Transformation initiatives/goals, [CONSTRAINTS] (capacity, budget), maturity assessment (PMO-04) if available

PROMPT

```
CONTEXT: Building the PMO/organizational transformation roadmap toward [BUSINESS_OUTCOME].
OBJECTIVE: Sequence transformation initiatives by organizational readiness and dependency,
avoiding parallel overload.
INPUTS: List of transformation initiatives, current maturity/readiness data, capacity/budget
constraints.
CONSTRAINTS: Do not schedule more concurrent major change initiatives than the organization's
demonstrated change-absorption capacity supports — state this capacity explicitly.
DECISION RULES: Sequence foundational initiatives (e.g., data/reporting infrastructure) before
initiatives that depend on them (e.g., advanced analytics-driven governance).
OUTPUT CONTRACT: Phased table [Phase | Initiative | Dependency | Readiness Check | Capacity
Consumed | Milestone].
VALIDATION: Confirm no phase exceeds the stated change-absorption capacity limit.
ESCALATION: Roadmap conflicts with concurrent business-as-usual demand route to executive sponsor
for a capacity/priority decision.
```

Output: Readiness- and capacity-checked transformation roadmap with named milestones.

Validate: Confirmed no phase exceeds the stated change-absorption capacity.

Next: Track via PMO-08 Executive Portfolio Report each cycle.

LEVEL L2 HUMAN GATE Yes

PMO-06 Benefits Realization Framework

Use when: Establishing or refreshing how the PMO tracks benefits realization across the portfolio.

Outcome: A benefits realization framework separating project completion from actual measured business benefit.

Inputs: Portfolio business cases, current benefits tracking practice, [BUSINESS_OUTCOME] targets

PROMPT

```
CONTEXT: Establishing the portfolio benefits realization framework across [PROGRAM]s.
OBJECTIVE: Define how benefits are tracked, measured and attributed, keeping project delivery
status and benefit realization as clearly separate signals.
INPUTS: Portfolio business cases with target benefits, current tracking practice (if any),
available measurement data sources.
CONSTRAINTS: The framework must require a named benefit owner (distinct from the project's
delivery owner) accountable for realization after go-live — delivery completion does not equal
benefit ownership discharge.
DECISION RULES: A benefit is only marked TRACKING if a specific measurement method and cadence is
defined; otherwise it remains UNTRACKED and is flagged as a framework gap.
OUTPUT CONTRACT: Sections — Framework Principles | Benefit Tracking Template [Benefit | Target |
Owner | Measurement Method | Cadence] | Portfolio Rollup Approach.
VALIDATION: Confirm every benefit has a named owner distinct from the delivery PM/TDM.
ESCALATION: Benefits with no owner or no measurement method route to sponsor for assignment
before the framework is adopted.
```

Output: Ownership-explicit benefits realization framework distinct from delivery tracking.

Validate: Every benefit confirmed to have a named owner distinct from the delivery lead.

Next: Apply per-program via TPM-09/PM-12 tracking.

LEVEL L2 HUMAN GATE Yes

PMO-07 Portfolio Risk Aggregation

Use when: Aggregating risk across the full portfolio to surface systemic, cross-program exposure.

Outcome: A portfolio risk view surfacing systemic risks that no single program-level view would reveal.

Inputs: Program-level risk registers (TPM-03 outputs), [RISK_TOLERANCE] at portfolio level

PROMPT

```
CONTEXT: Aggregating risk across the full portfolio of [PROGRAM]s against portfolio-level
[RISK_TOLERANCE].
OBJECTIVE: Identify systemic risks spanning multiple programs (e.g., shared vendor dependency,
common skill shortage, shared platform risk) invisible at single-program level.
INPUTS: Program-level risk registers/aggregations.
CONSTRAINTS: A risk is PORTFOLIO-SYSTEMIC only if it appears (independently or via shared root
cause) across 2+ programs — single-program risks stay at program level regardless of severity.
DECISION RULES: Elevate to portfolio risk register if systemic per the above rule, or if a single
program's risk realization would breach portfolio-level [RISK_TOLERANCE] (e.g., reputational or
regulatory exposure).
OUTPUT CONTRACT: Table [Risk | Type (Systemic/Elevated Single-Program) | Affected Programs |
Aggregate Score | Portfolio Response Owner].
VALIDATION: Systemic classification checked against the multi-program evidence rule.
ESCALATION: Portfolio-systemic risks route to PMO-01 Portfolio Health and executive briefing this
cycle.
```

Output: Portfolio-level systemic risk register distinguishing it from program-level risk.

Validate: Systemic classification checked against the multi-program evidence rule.

Next: Feed into PMO-01 Portfolio Health and executive briefing.

LEVEL L2 HUMAN GATE No

PMO-08 Executive Portfolio Report

Use when: Regular executive-level reporting across the full portfolio.

Outcome: An executive portfolio report anchored to business outcomes and decisions, not a status list per program.

Inputs: Portfolio health (PMO-01), risk aggregation (PMO-07), benefits status (PMO-06), [AUDIENCE], [DATE]

PROMPT

```
CONTEXT: Producing the [DATE] executive portfolio report for [AUDIENCE].
OBJECTIVE: Synthesize portfolio health, risk and benefits into an executive narrative anchored to
business outcomes, with clear decisions needed.
INPUTS: Portfolio health status, systemic risk register, benefits realization status.
CONSTRAINTS: Lead with business-outcome progress and decisions needed; per-program status detail
belongs in an appendix.
DECISION RULES: Any item requiring executive decision (funding, scope, priority trade-off) must
appear explicitly in a "Decisions Needed" section, distinct from status narrative.
OUTPUT CONTRACT: Sections — Portfolio Headline | Decisions Needed | Progress vs Business Outcomes
| Highest-Exposure Risks | Benefits Realization Snapshot | Appendix (per-program detail).
VALIDATION: Confirm every "Decisions Needed" item has at least two options and a named owner.
ESCALATION: N/A — this is the top-level reporting artifact; underlying escalations route via PMO-
07/TPM-11.
```

Output: Business-outcome-anchored executive portfolio report with clear decisions.

Validate: Every decision item confirmed to have options and a named owner.

Next: Distribute as pre-read before executive portfolio review.

LEVEL L1 HUMAN GATE Yes

PMO-09 Governance Automation Candidate Assessment

Use when: Identifying which PMO governance/reporting tasks could be automated to reduce manual reporting burden.

Outcome: A ranked list of governance automation candidates by manual effort saved and error-risk reduced.

Inputs: Current manual PMO reporting/governance tasks, frequency, time cost per cycle

PROMPT

```
CONTEXT: Assessing automation candidates among PMO governance/reporting tasks.
OBJECTIVE: Rank tasks for automation by manual effort saved and consistency/error-risk
improvement, not novelty of the automation approach.
INPUTS: Task list (e.g., status roll-up, RAG aggregation, KPI dashboard population), frequency,
time cost per cycle, current error/inconsistency rate if known.
CONSTRAINTS: Realistically estimate automation build/integration cost, including data source
reliability — do not assume clean, automatable source data without checking.
DECISION RULES: Prioritize HIGH if the task is high-frequency, rule-based (not requiring
judgment), and has a reliable structured data source; deprioritize tasks requiring significant
human judgment even if frequent.
OUTPUT CONTRACT: Table [Task | Frequency | Time Cost | Rule-Based? (Y/N) | Data Source
Reliability | Priority | Estimated Savings].
VALIDATION: Confirm HIGH priority items are confirmed rule-based and not judgment-dependent.
ESCALATION: N/A — feed into the PMO's own improvement backlog (PMO-05).
```

Output: Rule-based, ROI-ranked PMO automation candidate list.

Validate: High-priority items confirmed rule-based, not judgment-dependent.

Next: Feed into PMO-05 Transformation Roadmap as an improvement initiative.

LEVEL L1 HUMAN GATE No

PMO-10 AI Adoption Readiness Assessment

Use when: Assessing the PMO/portfolio's readiness to adopt AI-assisted delivery practices (including this prompt library).

Outcome: An honest AI adoption readiness assessment across data, process, skills and governance dimensions.

Inputs: Current tooling/data maturity, team AI literacy, governance posture, [CONSTRAINTS]

PROMPT

```
CONTEXT: Assessing AI adoption readiness for the PMO/portfolio at [CUSTOMER].
OBJECTIVE: Rate readiness across Data, Process, Skills and Governance dimensions with evidence,
identifying the actual blocking dimension.
INPUTS: Current data quality/availability for AI use cases, process standardization level, team
AI literacy/training status, existing AI governance policy.
CONSTRAINTS: Rate readiness from evidence (e.g., data actually accessible and clean, process
actually documented and consistent) — not from stated ambition or enthusiasm.
DECISION RULES: Flag the dimension with the LOWEST rating as the PRIMARY BLOCKER — adoption
efforts should target it first rather than spreading investment evenly.
OUTPUT CONTRACT: Table [Dimension | Readiness Rating | Evidence | Gap] + named Primary Blocker +
recommended first adoption step.
VALIDATION: Confirm the Primary Blocker is genuinely the lowest-rated dimension shown in the
table.
ESCALATION: Governance-dimension gaps (no AI usage policy, no human-review requirement defined)
route to AIX-12 AI Governance Framework before broader adoption proceeds.
```

Output: Dimension-rated AI readiness assessment with a named primary blocker.

Validate: Primary Blocker confirmed as the genuinely lowest-rated dimension.

Next: Governance gaps route to AIX-12 before broader adoption proceeds.

LEVEL L2 HUMAN GATE No

Agile & Product

38 Prompts Across 3 Practitioner Roles

Agile and product teams operate at the point where **customer needs, business priorities, requirements, team capacity, and delivery execution intersect**. I designed this section to help practitioners use AI to bring greater structure and clarity to that environment—without turning Agile into a collection of automated ceremonies or replacing the judgment required to make effective product decisions.

The **38 prompts** cover three complementary roles:

Role	Prompts	Primary Focus
Scrum Master / Agile Practitioner	**16**	Team effectiveness, Agile practices, impediments, flow, ceremonies, retrospectives and continuous improvement
Product Manager / Product Owner	**10**	Product priorities, outcomes, backlog decisions, value, stakeholder needs and product direction
Business Analyst	**12**	Requirements, analysis, clarification, traceability, process understanding and decision support

Connecting Product Intent to Delivery Execution

The three roles represent different perspectives on the same delivery system. **Product leadership** focuses on *what should be built and why*; **Business Analysis** helps establish *what is required and how the need should be understood*; while the **Scrum Master / Agile Practitioner** helps create the conditions for teams to deliver and continuously improve.

The prompts therefore focus on practical situations where ambiguity, competing priorities, incomplete information, or stakeholder expectations can slow delivery. AI can help structure the available information, expose gaps, compare alternatives, and prepare useful working outputs—allowing practitioners to focus their attention on the decisions and conversations that require human judgment.

I view AI in an Agile environment as an accelerator for the **feedback loop between intent, learning, and action**. It can help practitioners prepare for conversations, examine evidence from delivery and product data, identify patterns, challenge assumptions, and convert insights into actionable next steps.

For example, product and analysis work can establish clearer outcomes and requirements; those outputs can support backlog and delivery decisions; team-level insights can then feed retrospectives, prioritization, and subsequent product decisions.

The value is therefore not simply **“faster Agile documentation.”** The opportunity is to improve the **quality, consistency, and speed of the thinking that surrounds Agile delivery**.

AI-generated suggestions still require the context and judgment of the practitioner. Product priorities depend on business strategy and customer value; requirements require stakeholder interpretation; and Agile effectiveness depends on team dynamics that cannot be reduced to a prompt output.

Scrum Master / Agile Practitioner (SM)

Core agile team-health and flow prompts, each usable as-is or paired with a methodology adapter (SAFe, Kanban, XP, Lean/LeSS, Nexus/DASM) below.

Methodology Adapter Map

Methodology	Core Prompts Applied	Dedicated Adapter
Scrum	SM-01 – SM-11 apply natively	None needed
SAFe	SM-01 Iteration Health, SM-11 Dependencies	SM-12 PI / ART Sync Adapter
Kanban	SM-05 Flow Metrics	SM-13 Kanban Flow Adapter
XP	SM-06 Team Health, SM-07 Anti-Patterns	SM-14 Engineering Practice Adapter
Lean / LeSS	SM-04 Backlog Health, SM-11 Dependencies	SM-15 Large-Scale Adapter
Nexus / DASM	SM-09 Coaching Plan	SM-16 Tailoring Adapter
Hybrid Agile	Any core prompt	Blend adapters; document via SM-16

SM-01 Iteration Health Diagnostic

Use when: You need an evidence-based read on sprint/iteration health, not a gut-feel check-in.

Outcome: A diagnostic of iteration health across scope, flow, quality and predictability signals.

Inputs: [SOURCE_DATA] sprint board/burndown data, [TEAM] velocity history

PROMPT

```
CONTEXT: Diagnosing iteration health for <team> on [PROJECT], current iteration ending [DATE].
OBJECTIVE: Assess iteration health using burndown shape, scope change, carry-over and quality
signals.
INPUTS: Sprint board data, burndown/burnup, velocity history (last 4–6 iterations), scope-change
log.
CONSTRAINTS: Base findings only on board/metric data provided; do not speculate on team morale
without survey/retro evidence.
DECISION RULES: Flag AT RISK if burndown shows >20% remaining work with <20% time left, or if
mid-sprint scope change exceeds 15% of committed points.
OUTPUT CONTRACT: Sections — Health Rating (Green/Amber/Red) with evidence | Scope Stability |
Flow Signals | Quality Signals | Carry-over Trend (last 3 iterations).
VALIDATION: Every rating cites the specific metric threshold that triggered it.
ESCALATION: Red ratings 2 iterations running escalate to SM-07 Anti-Pattern Detection for root-
cause review.
```

Output: Iteration health rating with evidence-backed sub-signals.

Validate: Each rating traces to a specific metric threshold, not impression.

Next: Feed into retro (SM-03) and, if SAFe, the PI/ART adapter (SM-12).

LEVEL L1 HUMAN GATE No

SM-02 Impediment Analysis & Removal Plan

Use when: Impediments are logged but not resolving, or you need a prioritized removal plan.

Outcome: A prioritized impediment list with root cause and a specific removal action per item.

Inputs: Impediment log, [SOURCE_DATA] daily standup notes, days-open per impediment

PROMPT

```
CONTEXT: Reviewing open impediments for <team> on [PROJECT].
OBJECTIVE: Root-cause each impediment and define a specific removal action, prioritized by
delivery impact.
INPUTS: Impediment log with open dates, standup notes referencing the impediment.
CONSTRAINTS: Distinguish impediments the team can self-resolve from those requiring external
escalation.
DECISION RULES: ESCALATE if open >5 working days with no team-level resolution path; otherwise
assign to team action.
OUTPUT CONTRACT: Table [Impediment | Days Open | Root Cause | Removal Action | Owner | Escalate?
(Y/N)].
VALIDATION: Confirm days-open figures are recalculated from log dates, not carried from memory.
ESCALATION: Y-flagged items route to TDM-10-style escalation to the delivery lead this week.
```

Output: Prioritized impediment removal plan with escalation flags.

Validate: Days-open figures recalculated from the impediment log dates.

Next: Escalated items go to delivery/program leadership; track resolution in next standup.

LEVEL L1 HUMAN GATE No

SM-03 Retrospective Synthesis & Action Plan

Use when: You have raw retro input (sticky notes, survey, discussion notes) and need it converted to committed actions.

Outcome: A synthesized retro output with themed insights and a small set of committed, owned actions.

Inputs: Raw retro input/notes, [TEAM], prior retro actions and their completion status

PROMPT

```
CONTEXT: Synthesizing the retrospective for <team> on [PROJECT], iteration ending [DATE].
OBJECTIVE: Theme raw retro input into insights and convert them into a small number of committed,
trackable actions.
INPUTS: Raw retro notes/board export, status of actions committed in the prior retro.
CONSTRAINTS: Limit to a maximum of 3 new actions — more than 3 rarely gets done. Carry forward
incomplete prior actions rather than dropping them silently.
DECISION RULES: An item becomes an ACTION only if it is specific, owned, and completable within
one iteration; otherwise it becomes a PARKED THEME for later.
OUTPUT CONTRACT: Sections — Themes (grouped from raw input) | Prior Action Follow-up
(Done/Carried/Dropped with reason) | New Actions (max 3, each with owner) | Parked Themes.
VALIDATION: Confirm exactly ≤3 new actions and each has a named owner.
ESCALATION: Themes that recur 3+ retros running without resolution escalate to SM-07 Anti-Pattern
Detection.
```

Output: Themed retro synthesis with a max-3 committed action list.

Validate: Action count capped at 3 and each action has a named owner.

Next: Track actions in the team's board; recurring themes feed SM-07.

LEVEL L1 HUMAN GATE No

SM-04 Backlog Health Assessment

Use when: You suspect the backlog is bloated, stale, or poorly refined and need an objective assessment.

Outcome: A backlog health scorecard covering size, freshness, refinement depth and readiness.

Inputs: [SOURCE_DATA] backlog export, refinement history, definition of ready

PROMPT

```
CONTEXT: Assessing backlog health for [PROJECT] backlog owned by <team/PO>.
OBJECTIVE: Score backlog health on size, staleness, refinement depth and readiness against
Definition of Ready.
INPUTS: Full backlog export with last-updated dates and refinement status, Definition of Ready
criteria.
CONSTRAINTS: Score readiness only against a stated, explicit Definition of Ready — do not infer
an implicit standard.
DECISION RULES: Flag STALE if unchanged >60 days with no rationale. Flag OVERSIZED if backlog
exceeds ~3x expected iterations of runway at current velocity.
OUTPUT CONTRACT: Scorecard [Dimension | Rating | Evidence | Recommendation] covering Size,
Staleness, Refinement Depth, Ready-for-Sprint Count.
VALIDATION: Oversized calculation shown as (backlog item count ÷ average items completed per
iteration).
ESCALATION: If no Definition of Ready exists, flag DATA GAP and recommend defining one before
rating readiness.
```

Output: Backlog health scorecard with sizing math shown.

Validate: Oversized rating backed by the shown item-count-to-velocity calculation.

Next: Feed refinement gaps into next backlog refinement session.

LEVEL L1 HUMAN GATE No

SM-05 Flow Metrics Analysis

Use when: You need a data-driven view of how work actually flows through the team, beyond velocity.

Outcome: A flow analysis (cycle time, throughput, WIP, aging) identifying the actual bottleneck stage.

Inputs: [SOURCE_DATA] board/ticket data with stage timestamps, WIP limits if defined

PROMPT

```
CONTEXT: Analyzing flow for <team> on [PROJECT] over the last <period>.
OBJECTIVE: Identify the actual bottleneck stage in the workflow using cycle time, throughput and
WIP data.
INPUTS: Ticket-level stage transition timestamps, current WIP limits if any.
CONSTRAINTS: Identify the bottleneck as the stage with the highest average time-in-stage relative
to its WIP limit — do not assume it is "development" by default.
DECISION RULES: Flag a stage as BOTTLENECK if its average time-in-stage exceeds 1.5x the team's
overall average stage time.
OUTPUT CONTRACT: Table [Stage | Avg Time-in-Stage | WIP (current vs limit) | Throughput |
Bottleneck? (Y/N)] + one-paragraph explanation of the primary constraint.
VALIDATION: Bottleneck stage confirmed by the 1.5x threshold calculation, not assumption.
ESCALATION: If timestamp data is incomplete for a stage, flag DATA GAP for that stage rather than
estimating.
```

Output: Flow analysis identifying the calculated bottleneck stage.

Validate: Bottleneck flagged only where the 1.5x threshold calculation supports it.

Next: Pair with a Kanban adapter (SM-13) if using explicit WIP limits.

LEVEL L1 HUMAN GATE No

SM-06 Team Health Assessment

Use when: You want a structured, evidence-informed read on team health beyond delivery metrics.

Outcome: A team health assessment combining delivery signals with stated team sentiment (not inferred).

Inputs: Delivery metrics, team survey/sentiment data if collected, [TEAM]

PROMPT

```
CONTEXT: Assessing team health for <team> on [PROJECT].
OBJECTIVE: Combine delivery signals with actual collected sentiment data into a team health view.
INPUTS: Delivery metrics (SM-01/SM-05 outputs), team survey or sentiment-check results if
collected.
CONSTRAINTS: Do not infer psychological safety, morale or burnout from delivery metrics alone —
these require actual sentiment data; if absent, mark DATA GAP.
DECISION RULES: Flag a dimension for attention if delivery signal AND sentiment data both point
the same direction (e.g., declining velocity + declining sentiment score).
OUTPUT CONTRACT: Table [Dimension | Delivery Signal | Sentiment Data (or DATA GAP) | Combined
Read | Recommended Action].
VALIDATION: Confirm no psychological/morale claim is made without an actual sentiment data source
cited.
ESCALATION: Combined-signal concerns route to the SM/delivery lead for a direct team conversation
— do not action via written report alone.
```

Output: Team health view combining cited delivery and sentiment evidence.

Validate: No morale/psychological claim made without a cited sentiment data source.

Next: Discuss findings live with the team before broader distribution.

LEVEL L1 HUMAN GATE Yes

SM-07 Anti-Pattern Detection

Use when: Something feels systemically wrong with how the team works agile, and you need to name it precisely.

Outcome: A precise diagnosis of specific agile anti-patterns present, with evidence, not a vague critique.

Inputs: Recent iteration data (SM-01, SM-04, SM-05), retro history (SM-03), standup patterns

PROMPT

```
CONTEXT: Investigating potential agile anti-patterns for <team> on [PROJECT] based on recent
evidence.
OBJECTIVE: Identify specific, named anti-patterns (e.g., mini-waterfall, watermelon status, hero
culture, scope creep by stealth, zombie scrum) with supporting evidence.
INPUTS: Recent iteration health, backlog health, flow data, retro action history.
CONSTRAINTS: Name an anti-pattern only where at least two independent evidence points support it;
a single data point is INFERENCE, not a confirmed pattern.
DECISION RULES: CONFIRMED if 2+ independent evidence sources align; SUSPECTED if only 1 source;
classify accordingly.
OUTPUT CONTRACT: Table [Anti-Pattern | Status (Confirmed/Suspected) | Evidence Sources | Likely
Root Cause | Suggested Intervention].
VALIDATION: Every CONFIRMED entry cites 2+ distinct evidence sources by name.
ESCALATION: Confirmed anti-patterns tied to leadership behavior (e.g., hero culture driven by
management pressure) escalate beyond the team to delivery leadership.
```

Output: Named, evidence-graded anti-pattern diagnosis with suggested interventions.

Validate: CONFIRMED status requires 2+ distinct cited evidence sources.

Next: Leadership-driven patterns route to TDM/delivery lead conversation.

LEVEL L2 HUMAN GATE Yes

SM-08 Predictability & Forecast Analysis

Use when: Stakeholders want a delivery forecast and you need one grounded in actual variability, not a single average.

Outcome: A probabilistic forecast range using historical throughput variability, not a single-point guess.

Inputs: Historical velocity/throughput (min. 6 iterations), remaining scope size, [DATE]

PROMPT

```
CONTEXT: Forecasting delivery of remaining scope for <team>/[PROJECT] as of [DATE].
OBJECTIVE: Produce a probabilistic delivery forecast range using historical throughput
variability.
INPUTS: Historical velocity/throughput for at least 6 iterations, current remaining scope in
comparable units.
CONSTRAINTS: Do not forecast from fewer than 6 data points without flagging low confidence. Do
not present a single-point date as certain.
DECISION RULES: Use the range between the slowest and fastest historical throughput to compute a
best-case/worst-case forecast; state confidence as LOW if fewer than 6 data points are available.
OUTPUT CONTRACT: Forecast statement with best-case, likely-case and worst-case completion dates +
stated confidence level + key variability driver.
VALIDATION: Confirm the range is derived from actual historical min/max throughput, not an
assumed variance.
ESCALATION: LOW confidence forecasts should not be presented externally as commitments — flag for
internal use only until more data exists.
```

Output: Range-based delivery forecast with stated confidence level.

Validate: Forecast range derived from actual historical min/max, not assumption.

Next: Feed into PM-10/TDM-01 status reporting with confidence level stated.

LEVEL L1 HUMAN GATE Conditional

SM-09 Coaching Plan Builder

Use when: A team or individual needs a structured, evidence-based coaching plan rather than generic advice.

Outcome: A targeted coaching plan tied to specific observed behaviors and a defined success signal.

Inputs: Observed behavior/pattern (from SM-07 or direct observation), [TEAM] context, coaching goal

PROMPT

```
CONTEXT: Building a coaching plan for <team/individual> on [PROJECT] targeting observed pattern:
<describe>.
OBJECTIVE: Define a coaching approach with specific interventions and an observable success
signal.
INPUTS: The observed behavior/pattern with evidence, desired future state.
CONSTRAINTS: Ground every intervention in the specific observed pattern, not generic agile
coaching advice unrelated to the evidence.
DECISION RULES: Choose intervention type based on root cause: skill gap → training/pairing;
structural constraint → process change; motivation/trust → direct conversation, not process fix.
OUTPUT CONTRACT: Sections — Observed Pattern & Evidence | Likely Root Cause Category |
Intervention Plan (steps, timeframe) | Success Signal (specific, observable) | Review Date.
VALIDATION: Confirm intervention type matches the identified root cause category per the decision
rule.
ESCALATION: Interventions requiring management action (role change, structural team change) route
to delivery leadership.
```

Output: Targeted coaching plan with a root-cause-matched intervention and success signal.

Validate: Intervention type checked against the root-cause-category decision rule.

Next: Review against the success signal at the stated review date.

LEVEL L1 HUMAN GATE Conditional

SM-10 Release Readiness Assessment

Use when: Approaching a release and you need an objective go/no-go read across quality, scope and operational readiness.

Outcome: A structured go/no-go assessment with named blockers, not a subjective confidence statement.

Inputs: Release scope, defect data, test completion status, deployment/rollback plan status

PROMPT

```
CONTEXT: Assessing release readiness for <release name> on [PROJECT], target date [DATE].
OBJECTIVE: Produce a go/no-go recommendation across quality, scope completeness and operational readiness.
INPUTS: Release scope checklist, open defects by severity, test completion %, deployment/rollback plan status.
CONSTRAINTS: A GO recommendation requires zero open Critical/High defects unresolved and a validated rollback plan — no exceptions without explicit sponsor waiver.
DECISION RULES: NO-GO if any Critical defect open, or rollback plan untested. CONDITIONAL GO if only Medium/Low defects open, with named accepted risk.
OUTPUT CONTRACT: Sections — Recommendation (Go/No-Go/Conditional) | Blocking Items | Accepted Risks (if Conditional) | Rollback Plan Status | Sign-off Required From.
VALIDATION: Confirm the recommendation follows the stated decision rule exactly — no Critical-defect GO without an explicit waiver record.
ESCALATION: No-Go or Conditional-Go recommendations require named sign-off before release proceeds.
```

Output: Go/no-go release readiness recommendation with named blockers.

Validate: Recommendation logic checked against the critical-defect/rollback decision rule.

Next: Route to release approver for sign-off; log accepted risks in RAID.

LEVEL L1 HUMAN GATE Yes

SM-11 Cross-Team Dependency Management

Use when: Your team's delivery depends on another team's output and you need it actively tracked, not just noted.

Outcome: A tracked cross-team dependency view with status, risk and an explicit escalation trigger.

Inputs: Known cross-team dependencies, [SYSTEMS]/teams involved, required-by dates

PROMPT

```
CONTEXT: Managing cross-team dependencies for <team> on [PROJECT] involving <other teams/systems>.
OBJECTIVE: Track each dependency's status against the required-by date and flag emerging risk early.
INPUTS: Dependency list with providing team, required-by date, current status per last check-in.
CONSTRAINTS: Status must reflect the most recent confirmed check-in, not an assumption of "should be fine."
DECISION RULES: Flag AT RISK if the providing team's confirmed status is behind their own committed date by any amount, with no recovery plan stated.
OUTPUT CONTRACT: Table [Dependency | Providing Team | Required By | Last Confirmed Status | Flag | Next Check-in Date].
VALIDATION: Confirm each status entry has a dated check-in source, not "assumed on track."
ESCALATION: AT RISK dependencies within 2 iterations of required-by date escalate to both team leads jointly.
```

Output: Cross-team dependency tracker with dated status and early risk flags.

Validate: Every status entry sourced to a dated check-in, not assumption.

Next: Escalate jointly with the providing team's lead when flagged AT RISK.

LEVEL L1 HUMAN GATE No

SM-12 SAFe PI / ART Sync Adapter

Use when: Applying SM-01 (Iteration Health) or SM-11 (Dependencies) within a SAFe ART/PI context.

Outcome: PI-level health and dependency view using SAFe-specific artifacts (Features, WSJF, PI Objectives).

Inputs: PI Objectives, Program Board, Feature progress, ART sync notes

PROMPT

```
CONTEXT: Applying iteration/dependency health analysis at the ART level within [PROGRAM]'s
current PI.
OBJECTIVE: Adapt iteration health (SM-01) and cross-team dependency tracking (SM-11) to SAFe
artifacts — PI Objectives, Program Board, Features.
INPUTS: PI Objectives with business value scores, Program Board dependency links, per-team
Feature progress.
CONSTRAINTS: Use PI Objective business value scores as the priority lens, not raw story counts.
DECISION RULES: Flag a PI Objective AT RISK if its enabling Features are behind plan AND it sits
on the Program Board's cross-team dependency path.
OUTPUT CONTRACT: Table [PI Objective | Business Value | Enabling Features Status | Program Board
Dependency Risk | Flag] + ART-level narrative for PI System Demo.
VALIDATION: Confirm flags align cross-referenced Feature status with the Program Board dependency
links, not either alone.
ESCALATION: AT RISK PI Objectives route to the RTE for ART sync discussion.
```

Output: PI-level health view aligned to SAFe artifacts, ready for ART sync/System Demo.

Validate: Flags cross-reference both Feature status and Program Board dependencies.

Next: Present at ART Sync; unresolved risks to Inspect & Adapt.

LEVEL L1 HUMAN GATE No

SM-13 Kanban Flow Adapter

Use when: Applying SM-05 (Flow Metrics) within a Kanban system with explicit WIP limits and classes of service.

Outcome: A flow analysis using Kanban-specific levers: WIP limits, classes of service, and cumulative flow.

Inputs: Cumulative flow diagram data, WIP limits per column, classes of service if defined

PROMPT

```
CONTEXT: Applying flow analysis (SM-05) to a Kanban system for <team> on [PROJECT].
OBJECTIVE: Diagnose flow using cumulative flow diagram shape, WIP limit adherence and class-of-
service mix.
INPUTS: Cumulative flow diagram data, WIP limits per column, class-of-service tags if used.
CONSTRAINTS: Diagnose WIP limit breaches from actual column counts vs stated limits, not target
aspiration.
DECISION RULES: Flag a column as CONSTRAINED if it is at or above its WIP limit for >50% of the
analysis period. Flag Expedite-class overuse if Expedite items exceed 10% of throughput.
OUTPUT CONTRACT: Table [Column | WIP Limit | Time At/Above Limit % | Flag] + Class-of-Service mix
table + one bottleneck narrative.
VALIDATION: Confirm the ">50% of period" and "10% of throughput" calculations shown, not
asserted.
ESCALATION: Persistent WIP breaches with no policy change proposed route to team-level policy
review.
```

Output: Kanban-specific flow diagnostic with WIP and class-of-service analysis.

Validate: WIP-breach and Expedite-overuse percentages shown with the underlying calculation.

Next: Feed into a team WIP-limit or policy adjustment discussion.

LEVEL L1 HUMAN GATE No

SM-14 XP Engineering Practice Adapter

Use when: Applying SM-06/SM-07 (team health / anti-patterns) where engineering practice discipline is the likely lever.

Outcome: A diagnostic of XP practice adherence (TDD, pairing, CI, refactoring) tied to quality/flow outcomes.

Inputs: Test coverage trend, CI build data, pairing/mob frequency if tracked, defect escape data

PROMPT

```
CONTEXT: Applying team-health/anti-pattern analysis to engineering practice adherence for <team>
on [PROJECT].
OBJECTIVE: Assess whether XP practice gaps (TDD, pairing, CI hygiene, refactoring) are driving
observed quality or flow issues.
INPUTS: Test coverage trend, CI build success/duration data, pairing frequency if tracked, defect
escape rate.
CONSTRAINTS: Connect a practice gap to an outcome only where the data shows correlation (e.g.,
coverage decline preceding defect increase), not by assumption.
DECISION RULES: Flag CI HEALTH RISK if build failure rate >15% or average build time trending
upward 2+ periods. Flag TEST DEBT if coverage trend is declining for 2+ iterations.
OUTPUT CONTRACT: Table [Practice | Signal | Trend | Linked Outcome (if evidenced) |
Recommendation].
VALIDATION: Any "linked outcome" claim backed by a shown before/after data comparison.
ESCALATION: Sustained CI health risk blocking releases escalates to TL-10 CI/CD Pipeline
Assessment.
```

Output: XP practice-adherence diagnostic linked to quality/flow outcomes where evidenced.

Validate: Linked-outcome claims backed by a shown data comparison, not assumption.

Next: Escalate persistent CI issues to TL-10; feed test debt into TDM-09.

LEVEL L1 HUMAN GATE No

SM-15 Lean / LeSS Large-Scale Adapter

Use when: Applying SM-04/SM-11 (backlog health / dependencies) across multiple teams sharing one product backlog (LeSS/Lean at scale).

Outcome: A cross-team backlog and dependency view for a shared single-product-backlog structure.

Inputs: Shared product backlog, team allocation to backlog areas, cross-team refinement notes

PROMPT

```
CONTEXT: Applying backlog health and dependency analysis across multiple teams sharing one
product backlog under [PROGRAM] (LeSS/Lean-at-scale).
OBJECTIVE: Assess backlog health and dependency risk specific to a shared single-backlog, multi-
team structure.
INPUTS: Shared product backlog with team-of-origin/team-working tags, cross-team refinement
session notes.
CONSTRAINTS: Flag items only as cross-team dependent where two or more teams are explicitly
linked to the same backlog item or a shared component.
DECISION RULES: Flag COORDINATION RISK if 2+ teams are assigned overlapping backlog items without
a documented split/sequencing agreement.
OUTPUT CONTRACT: Table [Backlog Area | Teams Involved | Coordination Mechanism (or NONE) | Flag]
+ recommendation for overall/area refinement structure.
VALIDATION: Confirm every COORDINATION RISK entry names the specific overlapping backlog items.
ESCALATION: Persistent overlap with no coordination mechanism routes to the product group's
cross-team sync forum.
```

Output: Shared-backlog coordination risk view for multi-team LeSS/Lean structures.

Validate: Coordination-risk flags name the specific overlapping backlog items.

Next: Bring to overall/area refinement to establish a split agreement.

LEVEL L1 HUMAN GATE No

SM-16 Nexus / DASM Tailoring Adapter

Use when: Applying SM-09 (coaching) or selecting process approach where the team needs deliberate framework tailoring (Nexus integration or DASM lifecycle choice).

Outcome: A documented, rationale-based tailoring decision rather than a default framework choice.

Inputs: Team/product context, integration complexity, [CONSTRAINTS], candidate lifecycle/lens options

PROMPT

```
CONTEXT: Selecting or tailoring the agile approach (Nexus integration practices, or DASM
lifecycle/process choice) for <team(s)> on [PROJECT].
OBJECTIVE: Recommend a specific tailoring or lifecycle choice with explicit rationale tied to
context, not framework preference.
INPUTS: Team/product context (single team vs multi-team integration need), complexity and risk
profile, [CONSTRAINTS].
CONSTRAINTS: Do not recommend a framework/lifecycle by popularity — justify against the specific
context factors (integration frequency, risk, team distribution).
DECISION RULES: Recommend Nexus integration events if 3+ teams must integrate work into one
product at least every iteration. For DASM, choose Agile/Lean/Continuous Delivery lifecycle based
on team experience level and delivery cadence needs, per DASM's own selection factors.
OUTPUT CONTRACT: Sections — Context Summary | Options Considered | Recommended Approach with
rationale | Tailoring Notes (what to add/remove from the base framework) | Review Trigger (when
to revisit).
VALIDATION: Confirm the recommendation cites the specific context factor(s) that drove it, not a
generic best-practice statement.
ESCALATION: Tailoring that changes team structure or reporting lines requires delivery leadership
sign-off.
```

Output: Rationale-based framework tailoring/lifecycle recommendation with a review trigger.

Validate: Recommendation explicitly cites the context factors that drove the choice.

Next: Document in team working agreement; revisit at the stated review trigger.

LEVEL L1 HUMAN GATE Yes

Product Manager / Product Owner (PO)

Discovery, prioritization and product decision-making with explicit evidence and metrics discipline.

PO-01 Customer Problem Discovery Synthesis

Use when: You have raw discovery input (interviews, support tickets, survey data) and need it converted into problem statements.

Outcome: Synthesized, evidence-graded problem statements ranked by frequency and severity.

Inputs: [SOURCE_DATA] interview notes/tickets/survey data, [AUDIENCE] (customer segment)

PROMPT

```
CONTEXT: Synthesizing customer discovery input for [AUDIENCE] on [PRODUCT/PROJECT].
OBJECTIVE: Convert raw discovery input into ranked, evidence-graded problem statements.
INPUTS: Interview notes, support tickets, survey data — [SOURCE_DATA].
CONSTRAINTS: Every problem statement must cite the number of independent sources reporting it. Do
not merge distinct problems to inflate frequency.
DECISION RULES: Rank by (frequency × stated severity); frequency = count of independent sources,
severity from customer's own words where available, else DATA GAP.
OUTPUT CONTRACT: Table [Problem Statement | Frequency (source count) | Severity Evidence |
Segment | Rank].
VALIDATION: Recount frequency from source citations to confirm no inflation via merged problems.
ESCALATION: Problems with only 1 source are flagged LOW CONFIDENCE, not dropped — kept visible
but not prioritized.
```

Output: Ranked, source-counted problem statement list.

Validate: Frequency counts independently re-verified against cited sources.

Next: Top-ranked problems feed PO-02 Product Requirements Definition.

LEVEL L1 HUMAN GATE No

PO-02 Product Requirements Definition

Use when: A prioritized problem needs to become a defined requirement set before build.

Outcome: A requirements definition tied explicitly to the customer problem and success metric.

Inputs: Prioritized problem (PO-01), [BUSINESS_OUTCOME], [SUCCESS_CRITERIA]

PROMPT

```
CONTEXT: Defining requirements to address: <problem statement>, in service of [BUSINESS_OUTCOME].
OBJECTIVE: Define the requirement set with explicit success criteria, not just a feature
description.
INPUTS: The problem statement and evidence, target [SUCCESS_CRITERIA].
CONSTRAINTS: Every requirement must trace back to the stated problem — flag any requirement that
doesn't as SCOPE CREEP RISK.
DECISION RULES: Classify each requirement as MUST (blocks the core problem from being solved) or
SHOULD (enhances but not blocking).
OUTPUT CONTRACT: Sections — Problem Restated | Success Metric | Requirements (Must/Should, each
with rationale tracing to the problem) | Explicit Non-Goals.
VALIDATION: Every MUST requirement traces directly to a sentence in the problem statement.
ESCALATION: Requirements with no clear success metric route back for [SUCCESS_CRITERIA]
clarification before build.
```

Output: Problem-traced requirements set with must/should classification and non-goals.

Validate: Every MUST requirement traced to the specific problem statement.

Next: Feed MUST items into BA-04/BA-05 for story and acceptance criteria drafting.

LEVEL L1 HUMAN GATE Yes

PO-03 Prioritization Decision (RICE)

Use when: Comparing multiple backlog candidates and need a consistent, defensible priority order.

Outcome: A RICE-scored priority ranking with each score component shown, not just a final number.

Inputs: Candidate list, reach/impact estimates or basis for them, [THRESHOLD] for cut line

PROMPT

```
CONTEXT: Prioritizing candidate initiatives for [PROJECT]/[PRODUCT] using RICE.
OBJECTIVE: Score and rank candidates using Reach, Impact, Confidence, Effort with each component shown.
INPUTS: Candidate list with any available reach/impact evidence, effort estimates.
CONSTRAINTS: Confidence score must reflect actual evidence quality (High = validated data, Medium = directional signal, Low = opinion only) — do not default to High.
DECISION RULES: RICE score = (Reach × Impact × Confidence) ÷ Effort. Items below [THRESHOLD] are deprioritized, not dropped, unless explicitly killed.
OUTPUT CONTRACT: Table [Candidate | Reach | Impact | Confidence (with basis) | Effort | RICE Score | Rank].
VALIDATION: Recompute the top 3 scores manually to confirm formula was applied correctly.
ESCALATION: Items with Low confidence but high potential score route to PO-05 Experiment Design before committing resources.
```

Output: RICE-scored, ranked candidate list with visible score components.

Validate: Top-ranked scores manually recomputed to confirm formula accuracy.

Next: Low-confidence high-scorers go to PO-05 for validation before build.

LEVEL L1 HUMAN GATE No

PO-04 Roadmap Trade-off Analysis

Use when: Roadmap capacity is constrained and competing initiatives need an explicit trade-off decision.

Outcome: A trade-off analysis showing what's gained and given up for each roadmap sequencing option.

Inputs: Candidate initiatives with RICE/priority scores, [CONSTRAINTS] (capacity), [BUSINESS_OUTCOME]

PROMPT

```
CONTEXT: Sequencing the roadmap for [PRODUCT] under [CONSTRAINTS], targeting [BUSINESS_OUTCOME].
OBJECTIVE: Compare roadmap sequencing options and show explicit trade-offs, not a single "recommended" list.
INPUTS: Prioritized candidate list, available capacity by period.
CONSTRAINTS: Present at least 2 sequencing options with different trade-off profiles (e.g., "fast big bets" vs "steady incremental value").
DECISION RULES: For each option, state what value is captured earlier and what is deferred, and the deferred item's opportunity cost if known.
OUTPUT CONTRACT: Comparison table [Option | Sequence | Value Captured by Quarter | What's Deferred | Risk] + recommendation with rationale.
VALIDATION: Confirm each option's capacity allocation sums correctly against [CONSTRAINTS].
ESCALATION: Deferred items with committed customer expectations route to PRE/customer-engagement review before finalizing.
```

Output: Roadmap sequencing options with explicit value/deferral trade-offs.

Validate: Capacity allocation per option checked against stated constraints.

Next: Finalize with stakeholders; publish via itinerary-style roadmap view.

LEVEL L1 HUMAN GATE Yes

PO-05 Experiment Design & Hypothesis

Use when: A product idea carries real uncertainty and needs testing before full build investment.

Outcome: A testable hypothesis with a defined experiment, success threshold and decision rule.

Inputs: The idea/assumption to test, [SUCCESS_CRITERIA], available testing method/channel

PROMPT

```
CONTEXT: Designing an experiment to test the assumption: <state assumption> before committing to
full build.
OBJECTIVE: Define a falsifiable hypothesis, experiment design, and a pre-committed decision rule.
INPUTS: The assumption/idea, available testing channel (e.g., prototype, smoke test, A/B,
concierge test).
CONSTRAINTS: The hypothesis must be falsifiable and state a specific measurable threshold, not
"see how it goes."
DECISION RULES: Pre-commit the decision rule now: IF result ≥ [SUCCESS_CRITERIA] THEN proceed to
build. IF below THEN <specific alternative action>, decided before running the test.
OUTPUT CONTRACT: Sections — Hypothesis (falsifiable statement) | Experiment Design | Success
Threshold | Pre-committed Decision Rule | Sample Size/Duration Needed.
VALIDATION: Confirm the decision rule was written before any result is known (no post-hoc
threshold adjustment).
ESCALATION: Experiments requiring customer-facing exposure or brand risk require PRE/customer-
engagement or legal review before launch.
```

Output: Falsifiable experiment design with a pre-committed go/no-go decision rule.

Validate: Decision rule confirmed set before results, not adjusted after the fact.

Next: Run the experiment; result routes back to PO-03 prioritization.

LEVEL L2 HUMAN GATE Conditional

PO-06 Outcome Metrics Framework

Use when: You need to define what 'success' actually means for a product area beyond output/feature counts.

Outcome: An outcome metrics framework connecting product actions to measurable customer/business outcomes.

Inputs: [BUSINESS_OUTCOME], [PRODUCT] area, available data sources for measurement

PROMPT

```
CONTEXT: Defining outcome metrics for [PRODUCT] area in service of [BUSINESS_OUTCOME].
OBJECTIVE: Define a small set of outcome metrics (not output/vanity metrics) with clear
measurement sources.
INPUTS: Business outcome statement, available data/analytics sources.
CONSTRAINTS: Reject vanity metrics (e.g., "features shipped," "logins") unless explicitly tied to
a stated outcome with a causal rationale.
DECISION RULES: A metric qualifies as OUTCOME if it measures a change in customer behavior or
business result; otherwise classify as OUTPUT (track separately, do not treat as success).
OUTPUT CONTRACT: Table [Metric | Type (Outcome/Output) | Data Source | Current Baseline (or DATA
GAP) | Target] + North Star metric recommendation with rationale.
VALIDATION: Confirm each Outcome-classified metric passes the "change in behavior/result" test,
not just activity.
ESCALATION: Metrics with no available data source route to a measurement-instrumentation request
before adoption.
```

Output: Outcome-vs-output metrics framework with a recommended North Star.

Validate: Each outcome metric checked against the behavior-change test.

Next: Instrument missing data sources; review baseline quarterly.

LEVEL L1 HUMAN GATE No

PO-07 Value vs Effort Assessment

Use when: You need a fast, visual prioritization pass across a larger candidate set before deep RICE scoring.

Outcome: A value-vs-effort quadrant placement for rapid triage, with explicit placement rationale.

Inputs: Candidate list, rough value and effort signals

PROMPT

```
CONTEXT: Rapid triage of candidate initiatives for [PRODUCT] before deeper prioritization.
OBJECTIVE: Place each candidate on a value-vs-effort quadrant with a one-line rationale, to
filter for deeper RICE analysis.
INPUTS: Candidate list, any available value signal (customer demand, revenue potential) and
effort signal (rough sizing).
CONSTRAINTS: This is a triage pass, not a final decision — do not present quadrant placement as
committed prioritization.
DECISION RULES: Quick Win = high value/low effort; Big Bet = high value/high effort; Fill-in =
low value/low effort; Question Mark = low value/high effort (deprioritize or kill).
OUTPUT CONTRACT: Table [Candidate | Value Signal | Effort Signal | Quadrant | One-line Rationale]
grouped by quadrant.
VALIDATION: Confirm each quadrant assignment matches the stated value/effort signals, not
intuition alone.
ESCALATION: Quick Wins and Big Bets route to PO-03 RICE Prioritization for rigorous scoring.
```

Output: Value/effort quadrant triage with rationale, feeding deeper prioritization.

Validate: Quadrant placement checked against the stated value/effort signals.

Next: Quick Wins and Big Bets proceed to PO-03 RICE scoring.

LEVEL L1 HUMAN GATE No

PO-08 WSJF Scoring Model

Use when: Operating in a SAFe/scaled context and need Weighted Shortest Job First scoring for economic prioritization.

Outcome: A WSJF-scored priority list using the standard cost-of-delay-over-duration formula, transparently.

Inputs: Candidate features, user/business value, time criticality, risk reduction/opportunity, job size estimates

PROMPT

```
CONTEXT: Prioritizing features/epics for [PROGRAM] using WSJF.
OBJECTIVE: Score candidates using Cost of Delay (User-Business Value + Time Criticality + Risk
Reduction-Opportunity Enablement) ÷ Job Size.
INPUTS: Candidate list with relative estimates (e.g., Fibonacci) for each WSJF component.
CONSTRAINTS: Use relative sizing (not absolute units) consistently across all candidates being
compared in the same session.
DECISION RULES: WSJF = (User-Business Value + Time Criticality + Risk Reduction/Opportunity
Enablement) ÷ Job Size. Higher score = higher priority.
OUTPUT CONTRACT: Table [Candidate | User-Business Value | Time Criticality | RR/OE | Cost of
Delay | Job Size | WSJF Score | Rank].
VALIDATION: Recompute the top-ranked item's formula manually to confirm correct arithmetic.
ESCALATION: Ties or near-ties at the cut line route to a facilitated relative-estimation session
rather than an arbitrary tiebreak.
```

Output: WSJF-scored, ranked feature/epic priority list.

Validate: Top-ranked WSJF score manually recomputed for accuracy.

Next: Feed into PI planning / roadmap sequencing (see SM-12 SAFe adapter).

LEVEL L1 HUMAN GATE No

PO-09 MVP Scope Definition

Use when: You need to define the smallest viable release that still tests or delivers the core value proposition.

Outcome: A defensible MVP scope with explicit cut lines, distinct from a 'phase 1' feature dump.

Inputs: Product requirements (PO-02), the core value hypothesis, [CONSTRAINTS]

PROMPT

```
CONTEXT: Defining MVP scope for <initiative> on [PRODUCT], core value hypothesis: <state it>.
OBJECTIVE: Define the smallest scope that validates or delivers the core value hypothesis, with
explicit exclusions.
INPUTS: Full requirements set (PO-02), the core value hypothesis being tested/delivered,
constraints (time, budget).
CONSTRAINTS: Every included item must be necessary to test or deliver the core hypothesis — "nice
to have" items are explicitly excluded, not silently trimmed.
DECISION RULES: Include an item ONLY IF removing it would prevent the core hypothesis from being
validly tested/delivered; otherwise move to Explicit Non-Goals for a later phase.
OUTPUT CONTRACT: Sections — Core Value Hypothesis | MVP Scope (each item with why it's necessary)
| Explicit Non-Goals / Phase 2 | Risks of the Reduced Scope.
VALIDATION: Every MVP item's "why necessary" statement is tested against the hypothesis, not
general usefulness.
ESCALATION: Stakeholder pushback to add "nice to have" items routes to PO-04 Trade-off Analysis
rather than silent scope growth.
```

Output: Hypothesis-tested MVP scope with explicit non-goals and reduced-scope risks.

Validate: Each included item's necessity checked against the core hypothesis.

Next: Reject scope creep via PO-04 Trade-off Analysis, not ad hoc addition.

LEVEL L1 HUMAN GATE Yes

PO-10 Product Decision Brief

Use when: A significant product decision needs to be documented and communicated with clear rationale.

Outcome: A concise, evidence-backed product decision record for stakeholder alignment and future reference.

Inputs: The decision made, evidence considered, [SUCCESS_CRITERIA] for evaluating it later

PROMPT

```
CONTEXT: Documenting a product decision on [PRODUCT]: <state decision>.
OBJECTIVE: Produce a decision record capturing what was decided, why, and how it will be
evaluated later.
INPUTS: The decision, evidence/options considered, [SUCCESS_CRITERIA] to judge it against.
CONSTRAINTS: State the decision and alternatives considered — do not present the chosen option as
the only one evaluated if others existed.
DECISION RULES: Include a specific "we will know this was right/wrong if..." statement — this is
mandatory.
OUTPUT CONTRACT: Sections — Decision | Context/Problem | Options Considered | Rationale | Success
Criteria for Review | Review Date.
VALIDATION: Confirm the review criteria are specific and measurable, not "see how it feels."
ESCALATION: Decisions affecting committed roadmap dates route to stakeholders for awareness
before publishing.
```

Output: Structured product decision record with a measurable future review trigger.

Validate: Review criteria confirmed specific and measurable.

Next: Revisit at the stated review date; feed outcome into PO-06 metrics.

LEVEL L1 HUMAN GATE Yes

Business Analyst (BA)

Requirements discovery, decomposition and traceability with explicit fact/assumption discipline.

BA-01 Requirements Discovery Plan

Use when: Starting elicitation on a new initiative and need a structured discovery approach, not ad hoc questions.

Outcome: A discovery plan mapping stakeholders to elicitation techniques and the specific gaps each will close.

Inputs: [OBJECTIVE], known stakeholders, [SOURCE_DATA] existing documentation

PROMPT

```
CONTEXT: Planning requirements discovery for [OBJECTIVE] on [PROJECT].
OBJECTIVE: Design a discovery plan that matches elicitation technique to stakeholder type and identifies known information gaps up front.
INPUTS: Objective statement, stakeholder list with roles, existing documentation already available.
CONSTRAINTS: Do not plan a workshop for information already available in [SOURCE_DATA] — identify what's already known vs what requires elicitation.
DECISION RULES: Use workshops for cross-functional process/rule discovery; 1:1 interviews for sensitive or political topics; document analysis first for anything with existing artifacts.
OUTPUT CONTRACT: Table [Stakeholder/Group | Known Gap to Close | Technique | Est. Duration | Pre-read Needed] + list of gaps already closed by existing documentation.
VALIDATION: Confirm no planned session duplicates information already available in reviewed documentation.
ESCALATION: Stakeholders who decline or are unavailable for a required session are flagged as a discovery risk.
```

Output: Discovery plan matched to stakeholder type with identified pre-existing gaps closed.

Validate: No session planned duplicates already-available documented information.

Next: Execute sessions; feed raw output into BA-02 Requirements Decomposition.

LEVEL L1 HUMAN GATE No

BA-02 Requirements Decomposition

Use when: A high-level requirement or epic needs to be broken into implementable, testable components.

Outcome: Decomposed requirements at a testable level of granularity, each traceable to the parent.

Inputs: High-level requirement/epic, [SOURCE_DATA] supporting discovery notes

PROMPT

```
CONTEXT: Decomposing the high-level requirement: <state requirement/epic> for [PROJECT].
OBJECTIVE: Break the requirement into child requirements at a level testable independently, each traceable to the parent.
INPUTS: Parent requirement statement, discovery notes/evidence.
CONSTRAINTS: Each child requirement must be independently verifiable (pass/fail testable) — reject compound requirements bundling multiple conditions.
DECISION RULES: If a requirement contains "and"/"or" joining distinct testable conditions, split it into separate child requirements.
OUTPUT CONTRACT: Table [Child Requirement ID | Statement | Parent Ref | Testable? (Y/N) | Source].
VALIDATION: Spot-check 3 child requirements for compound-condition violations.
ESCALATION: Requirements with no traceable source are flagged DATA GAP, not invented to fill the decomposition.
```

Output: Decomposed, independently-testable child requirements with parent traceability.

Validate: Sample-checked for compound conditions that should have been split further.

Next: Feed into BA-04 User Stories and BA-11 Traceability Matrix.

LEVEL L1 HUMAN GATE No

BA-03 Ambiguity & Conflict Resolution

Use when: Requirements from different sources conflict or are ambiguous and need resolution before build.

Outcome: A resolved position on each ambiguity/conflict, or an explicit escalation where resolution needs a decision-maker.

Inputs: Conflicting/ambiguous requirement statements with sources

PROMPT

```
CONTEXT: Resolving ambiguity/conflict in requirements for [PROJECT]: <describe the conflicting
statements and sources>.
OBJECTIVE: Determine whether the conflict is resolvable from existing evidence or requires
stakeholder decision.
INPUTS: The conflicting statements with their sources (who said what, when).
CONSTRAINTS: Do not silently pick one interpretation — show both readings and the evidence for
each.
DECISION RULES: RESOLVABLE if one source has clear authority (e.g., signed-off business rule
document) over the other; otherwise ESCALATE for a decision.
OUTPUT CONTRACT: Table [Conflict | Reading A (source) | Reading B (source) | Resolution or
ESCALATE | Rationale].
VALIDATION: Confirm authority-based resolutions cite the specific governing document/decision.
ESCALATION: ESCALATE items route to the named business owner with a specific question to answer,
not a general "please clarify."
```

Output: Resolved or explicitly escalated requirement conflicts with cited rationale.

Validate: Authority-based resolutions cite the specific governing source.

Next: Escalated items get a specific question routed to the business owner.

LEVEL L1 HUMAN GATE Conditional

BA-04 User Story Writing

Use when: Converting a defined requirement into build-ready user stories.

Outcome: Well-formed user stories with clear value statements, sized for a single iteration.

Inputs: Decomposed requirement (BA-02), [AUDIENCE]/persona, [SUCCESS_CRITERIA]

PROMPT

```
CONTEXT: Writing user stories for requirement: <state requirement> for persona [AUDIENCE] on
[PROJECT].
OBJECTIVE: Produce well-formed, independently valuable user stories sized for a single iteration.
INPUTS: The requirement statement, target persona/user type, business value context.
CONSTRAINTS: Each story must deliver standalone user value — reject technical-task-only stories
disguised as user stories.
DECISION RULES: If a story cannot be completed within a single iteration at typical team
velocity, flag for further splitting.
OUTPUT CONTRACT: List of stories in "As a [persona], I want [capability], so that [value]" form,
each tagged [Independently Valuable: Y/N] and [Estimated Fit: Fits/Needs Split].
VALIDATION: Confirm each story's "so that" clause states genuine user/business value, not a
restated task.
ESCALATION: Stories needing splits route back through BA-02 decomposition logic.
```

Output: Iteration-sized user stories with value statements and fit assessment.

Validate: Each story's value clause checked for genuine benefit, not restated task.

Next: Feed directly into BA-05 Acceptance Criteria Definition.

LEVEL L1 HUMAN GATE No

BA-05 Acceptance Criteria Definition

Use when: A user story needs precise, testable acceptance criteria before development starts.

Outcome: Testable acceptance criteria in Given/When/Then form covering happy path and key edge cases.

Inputs: User story (BA-04), known business rules, [CONSTRAINTS]

PROMPT

```
CONTEXT: Defining acceptance criteria for story: <state story> on [PROJECT].
OBJECTIVE: Produce acceptance criteria covering the happy path and material edge cases, each
independently testable.
INPUTS: The user story, related business rules, known constraints/exceptions.
CONSTRAINTS: Do not invent business rules not evidenced in [SOURCE_DATA] — if an edge case's
expected behavior is unknown, flag DATA GAP rather than assume.
DECISION RULES: Include an edge case as a criterion only if it represents a materially different
system behavior; do not pad with trivial restatements.
OUTPUT CONTRACT: Given/When/Then criteria list, tagged [Happy Path] or [Edge Case], each with a
testability check (pass/fail determinable).
VALIDATION: Confirm every criterion is independently pass/fail testable, not a vague quality
statement.
ESCALATION: Edge cases marked DATA GAP route to the product owner/business rule owner for a
ruling before dev starts.
```

Output: Given/When/Then acceptance criteria set, tagged and testability-checked.

Validate: Every criterion confirmed independently testable as pass/fail.

Next: Feed into QA-02 Test Plan Generator and BA-11 Traceability Matrix.

LEVEL L1 HUMAN GATE No

BA-06 Business Rules Extraction

Use when: Business rules are scattered across documents/conversations and need to be extracted into one governed set.

Outcome: A consolidated, sourced business rules catalog free of contradictions.

Inputs: [SOURCE_DATA] policy docs, process docs, SME interview notes

PROMPT

```
CONTEXT: Extracting business rules governing <process/domain> for [PROJECT] from available source
material.
OBJECTIVE: Consolidate business rules into a single catalog, each traced to source, with
contradictions flagged.
INPUTS: Policy documents, process documentation, SME interview notes.
CONSTRAINTS: Extract only rules explicitly stated or directly implied by a specific passage — no
invented rules to "fill gaps."
DECISION RULES: Flag CONTRADICTION if two sourced rules govern the same condition with different
outcomes.
OUTPUT CONTRACT: Table [Rule ID | Rule Statement | Condition | Outcome | Source | Contradiction
Flag].
VALIDATION: Spot-check that each rule statement matches its cited source passage's actual
content.
ESCALATION: Contradictions route to the process/policy owner for a definitive ruling.
```

Output: Sourced, contradiction-flagged business rules catalog.

Validate: Sampled rules confirmed to match their cited source passages.

Next: Feed rules into BA-05 Acceptance Criteria and BA-08 Process Mapping.

LEVEL L1 HUMAN GATE No

BA-07 Non-Functional Requirements Definition

Use when: NFRs are vague or missing and need to be made specific and testable.

Outcome: Specific, measurable NFRs by category, each with a defined verification method.

Inputs: [SYSTEMS] context, [SUCCESS_CRITERIA], known regulatory/contractual constraints

PROMPT

```
CONTEXT: Defining non-functional requirements for [SYSTEMS] on [PROJECT].
OBJECTIVE: Convert vague NFR statements (e.g., "fast," "secure," "scalable") into specific,
measurable, testable requirements.
INPUTS: System context, business [SUCCESS_CRITERIA], known regulatory or contractual obligations.
CONSTRAINTS: Reject any NFR without a measurable threshold and a stated verification method —
"should be fast" is not acceptable output.
DECISION RULES: Categorize under standard NFR categories (Performance, Availability, Security,
Scalability, Usability, Compliance, Maintainability) and require a metric + method for each.
OUTPUT CONTRACT: Table [Category | Requirement (with metric/threshold) | Verification Method |
Source/Driver (regulatory, contractual, or business need)].
VALIDATION: Confirm every requirement includes both a numeric/measurable threshold and a named
verification method.
ESCALATION: Categories with no available threshold data route to Solution/Cloud Architect for
engineering input.
```

Output: Measurable, verifiable NFR set organized by standard category.

Validate: Every NFR carries both a numeric threshold and a named verification method.

Next: Hand to SA-03 NFR Definition & Validation for architecture-level validation.

LEVEL L1 HUMAN GATE No

BA-08 Process Mapping (As-Is / To-Be)

Use when: You need to document current process reality and design the target state, distinctly.

Outcome: A clear as-is process map and a to-be design with explicit deltas called out.

Inputs: Process walkthrough notes/observations, [BUSINESS_OUTCOME] for the to-be state

PROMPT

```
CONTEXT: Mapping the <process name> process for [PROJECT] — current (as-is) and target (to-be)
states.
OBJECTIVE: Document the as-is process from direct evidence, then design a to-be process aligned
to [BUSINESS_OUTCOME], with deltas explicit.
INPUTS: Process walkthrough notes, system logs/observations if available, target outcome
statement.
CONSTRAINTS: The as-is map must reflect what actually happens (including workarounds), not the
documented "official" process if they differ — note both if they diverge.
DECISION RULES: Every to-be step that differs from as-is must state the specific problem it
solves.
OUTPUT CONTRACT: Two swimlane-style step tables [As-Is: Step | Actor | System | Pain Point] and
[To-Be: Step | Actor | System | Problem Solved] + a delta summary.
VALIDATION: Confirm every to-be delta traces to a named as-is pain point.
ESCALATION: To-be designs requiring new system capability route to SA/Engineering for feasibility
input.
```

Output: As-is/to-be process maps with an explicit, justified delta summary.

Validate: Every to-be change traced to a specific as-is pain point.

Next: Feasibility-check new capability needs with Solution Architect (SA-10).

LEVEL L2 HUMAN GATE No

BA-09 Gap Analysis

Use when: You need to identify the specific gap between current and desired capability/state.

Outcome: A structured gap analysis with each gap sized and prioritized, not a vague list of differences.

Inputs: Current state description, [BUSINESS_OUTCOME]/target state, [SUCCESS_CRITERIA]

PROMPT

```
CONTEXT: Performing gap analysis between current state and target [BUSINESS_OUTCOME] for
[PROJECT].
OBJECTIVE: Identify specific, sized capability gaps and prioritize them by impact on reaching the
target state.
INPUTS: Current state description/data, target state definition, success criteria.
CONSTRAINTS: Each gap must be stated as a specific, closable difference (capability, data,
process, or system), not a general observation.
DECISION RULES: Prioritize gaps by (impact on reaching [BUSINESS_OUTCOME] × ease of closing) —
show both factors, don't just assert priority.
OUTPUT CONTRACT: Table [Gap | Current State | Target State | Impact if Unclosed | Closing Effort
(rough) | Priority].
VALIDATION: Confirm each gap statement names a specific, closable difference rather than a vague
concern.
ESCALATION: High-impact/high-effort gaps route to a dedicated feasibility or investment
discussion.
```

Output: Prioritized, sized gap analysis against the target state.

Validate: Each gap confirmed specific and closable, not a vague observation.

Next: High-priority gaps feed roadmap/requirements planning (PO-04, BA-01).

LEVEL L1 HUMAN GATE No

BA-10 Stakeholder Analysis (BA View)

Use when: You need to understand who has authority over which requirements decisions before eliciting or resolving conflicts.

Outcome: A requirements-decision-focused stakeholder map: who can approve, who must be consulted, who is informed.

Inputs: Known stakeholders, [OBJECTIVE], org context

PROMPT

```
CONTEXT: Mapping requirements decision authority for [OBJECTIVE] on [PROJECT].
OBJECTIVE: Identify who has approval authority, who must be consulted, and who is informed-only
for requirements decisions.
INPUTS: Stakeholder list, known org roles/authority, prior decision patterns if known.
CONSTRAINTS: Do not assume approval authority from seniority alone — base it on documented
ownership/governance where possible; else DATA GAP.
DECISION RULES: Classify each stakeholder as Approve / Consult / Inform per requirements-decision
RACI logic.
OUTPUT CONTRACT: Table [Stakeholder | Role | Requirements Authority (Approve/Consult/Inform) |
Basis for Classification | Notes].
VALIDATION: Confirm Approve-classified stakeholders have a stated governance basis, not assumed
seniority.
ESCALATION: Unclear approval authority routes to the project sponsor for an explicit RACI ruling
before requirements sign-off.
```

Output: Requirements-decision RACI map for stakeholders.

Validate: Approve classifications backed by a stated governance basis.

Next: Use this RACI when routing BA-03 conflict escalations.

LEVEL L1 HUMAN GATE No

BA-11 Requirements Traceability Matrix

Use when: You need end-to-end traceability from business need through requirement to test, for audit/governance.

Outcome: A traceability matrix linking business need → requirement → design → test, with orphan items flagged.

Inputs: Business objectives, requirements (BA-02), design artifacts, test cases

PROMPT

```
CONTEXT: Building the requirements traceability matrix for [PROJECT] against [OBJECTIVE].
OBJECTIVE: Link every requirement to its originating business need and its downstream design/test coverage; flag orphans in either direction.
INPUTS: Business objectives/need statements, requirements list, design/solution references, test case list.
CONSTRAINTS: A link is valid only if it is explicitly documented (e.g., a test case that states which requirement it verifies) — do not infer links by topic similarity alone.
DECISION RULES: Flag ORPHAN REQUIREMENT if no business need maps to it. Flag UNTESTED if no test case references it.
OUTPUT CONTRACT: Table [Business Need | Requirement ID | Design Ref | Test Case ID | Status (Complete/Orphan/Untested)].
VALIDATION: Spot-check 3 "Complete" rows to confirm the links are explicitly documented, not inferred.
ESCALATION: Orphan requirements route back to BA-01/BA-03 to confirm origin or remove from scope; untested requirements route to QA-03.
```

Output: End-to-end traceability matrix with orphan/untested flags.

Validate: Sampled complete rows confirmed to have explicitly documented links.

Next: Untested items route to QA-03 Requirements-to-Test Traceability.

LEVEL L2 HUMAN GATE No

BA-12 Impact Analysis

Use when: A proposed change needs its full downstream impact assessed before approval.

Outcome: A requirements-level impact analysis covering affected requirements, processes, systems and stakeholders.

Inputs: Proposed change description, requirements traceability matrix (BA-11), [SYSTEMS] map

PROMPT

```
CONTEXT: Assessing the requirements-level impact of a proposed change on [PROJECT]: <describe change>.
OBJECTIVE: Identify all requirements, processes, systems and stakeholders affected by the change, using traceability data.
INPUTS: Change description, traceability matrix, systems/process maps.
CONSTRAINTS: Trace impact only through documented links (traceability matrix, process maps) — do not speculate on indirect impact without a traceable path.
DECISION RULES: Classify impact as DIRECT (explicit traceable link) or POTENTIAL (shared system/process, needs verification).
OUTPUT CONTRACT: Table [Affected Item | Type (Requirement/Process/System/Stakeholder) | Impact Classification | Evidence Path | Recommended Verification Action].
VALIDATION: Every DIRECT classification cites the specific traceability link used.
ESCALATION: High-volume or high-criticality impact routes to PM-06 Change Request Impact Analysis for formal change control.
```

Output: Traceability-grounded impact analysis distinguishing direct from potential effects.

Validate: Direct impact classifications cite the specific traceability link.

Next: Feed into PM-06 for formal change control decision.

LEVEL L2 HUMAN GATE No

Engineering & Architecture

66 Prompts Across 7 Practitioner Roles

Engineering and architecture sit at the intersection of **business needs, technical complexity, security, data, infrastructure, and enterprise constraints**. I designed this section to help technical practitioners use AI to accelerate analysis, explore alternatives, surface risks, and structure architecture and engineering decisions—while keeping technical accountability with the practitioner.

Role	Prompts	Core Focus
Technical / Engineering Lead	**12**	Technical execution, engineering decisions & problem-solving
Solution Architect	**12**	Solution design, integration & architecture trade-offs
Cloud Architect	**10**	Cloud design, migration, scalability, resilience & cost
Infrastructure Architect	**8**	Infrastructure, environments, capacity & reliability
Enterprise Architect	**8**	Technology strategy, standards & target architecture
Security Architect	**8**	Threats, controls, security risk & secure design
Data Architect	**8**	Data architecture, integration, governance & quality

From Complexity to Better Technical Decisions

The prompts focus on situations where technical teams must evaluate **constraints, dependencies, design alternatives, non-functional requirements, risks, and downstream impacts**. Rather than asking AI to make architecture decisions autonomously, I use it as a **technical thinking accelerator**—helping practitioners structure the problem, challenge options, identify gaps, and prepare decision-ready artifacts.

The seven roles provide complementary perspectives across the technology ecosystem: **engineering → solution → cloud → infrastructure → enterprise → security → data**. This makes the prompts useful not only within individual roles, but also as inputs to **cross-functional architecture and engineering conversations**.

The Principle

Use AI to accelerate technical reasoning—not to outsource technical accountability.

Every AI-assisted technical output still needs appropriate validation against **source evidence, constraints, architecture standards, security requirements, operational realities, and production impact** before it becomes an engineering or architecture decision.

Technical Lead / Engineering Lead (TL)

Design, code quality, technical risk and engineering decision support grounded in evidence, not opinion.

TL-01 Technical Design Review

Use when: A design document needs structured, evidence-based critique before implementation begins.

Outcome: A structured design review flagging risks, gaps and alternatives, not a rubber-stamp approval.

Inputs: Design document, [SYSTEMS] context, [SUCCESS_CRITERIA]/NFRs

PROMPT

```
CONTEXT: Reviewing the technical design for <component/feature> on [PROJECT] against stated
NFRs/[SUCCESS_CRITERIA].
OBJECTIVE: Identify design risks, unaddressed NFRs and viable alternatives before implementation
proceeds.
INPUTS: Design document, relevant NFRs, [SYSTEMS] this design integrates with.
CONSTRAINTS: Flag any NFR (performance, security, scalability) not explicitly addressed in the
design — do not assume it was considered.
DECISION RULES: BLOCKING if the design cannot meet a stated NFR or introduces a single point of
failure with no mitigation; NON-BLOCKING for style/preference issues.
OUTPUT CONTRACT: Table [Concern | Severity (Blocking/Non-blocking) | Rationale | Suggested
Alternative] + overall recommendation (Approve/Approve with changes/Reject).
VALIDATION: Confirm every BLOCKING item ties to a specific unaddressed NFR or concrete failure
mode, not a style preference.
ESCALATION: BLOCKING items require design owner response before implementation sign-off.
```

Output: Severity-graded design review with recommendation and suggested alternatives.

Validate: Blocking items confirmed tied to specific NFRs or failure modes.

Next: Blocking items resolved before sign-off; approved design feeds estimation (TL-11).

LEVEL L1 HUMAN GATE Yes

TL-02 Code Review Assistant

Use when: You need a consistent, structured second pass on a code change beyond style/lint checks.

Outcome: A structured code review flagging correctness, security and maintainability issues with severity.

Inputs: Code diff/PR content, [CONSTRAINTS] (coding standards), context on the change's purpose

PROMPT

```
CONTEXT: Reviewing a code change for <component> on [PROJECT]. Purpose of change: <state it>.
OBJECTIVE: Identify correctness, security, performance and maintainability issues, each with
severity and rationale.
INPUTS: The code diff/PR, stated coding standards, purpose/intent of the change.
CONSTRAINTS: Do not flag pure style preferences unless they violate a documented standard — focus
on correctness, security and maintainability substance.
DECISION RULES: BLOCKING for correctness bugs, security vulnerabilities, or missing error
handling on failure paths; SUGGESTION for maintainability improvements.
OUTPUT CONTRACT: Table [File/Line Ref | Issue | Severity | Rationale | Suggested Fix] + summary
verdict (Approve/Request Changes).
VALIDATION: Confirm every BLOCKING item has a concrete failure scenario described, not a
hypothetical "could be an issue."
ESCALATION: Security-classified BLOCKING issues route to SEC review before merge.
```

Output: Severity-graded code review with concrete fix suggestions.

Validate: Blocking items each describe a concrete failure scenario.

Next: Security-flagged items route to SEC-05 Security Controls Gap Analysis.

LEVEL L1 HUMAN GATE Yes

TL-03 Technical Debt Assessment

Use when: You need to catalog and prioritize technical debt for engineering planning conversations.

Outcome: A prioritized technical debt inventory scored by delivery/operational risk, not just code smell count.

Inputs: [SYSTEMS] codebase context, known pain points, incident/defect history if available

PROMPT

```
CONTEXT: Assessing technical debt in [SYSTEMS] for [PROJECT] engineering planning.
OBJECTIVE: Catalog debt items and score each by operational/delivery risk, not just code-quality
aesthetics.
INPUTS: Known debt items/pain points, incident or defect history connected to affected areas,
upcoming roadmap touching those areas.
CONSTRAINTS: Score risk based on evidence (incident frequency, defect rate, delivery friction)
where available; otherwise mark confidence LOW.
DECISION RULES: HIGH risk if debt has caused 2+ incidents/defects in the last period OR blocks an
upcoming roadmap item; else MEDIUM or LOW.
OUTPUT CONTRACT: Table [Debt Item | Affected Area | Evidence | Risk Rating | Confidence |
Remediation Effort (rough) | Recommendation (Fix now/Schedule/Accept)].
VALIDATION: HIGH ratings confirmed against the stated incident/defect/roadmap-blocking evidence.
ESCALATION: HIGH-risk unfunded items route to TDM-09 Technical Debt Delivery Impact for business
framing.
```

Output: Risk-scored, evidence-based technical debt inventory.

Validate: HIGH ratings checked against cited incident/defect/roadmap evidence.

Next: Unfunded HIGH items feed TDM-09 for executive framing.

LEVEL L1 HUMAN GATE No

TL-04 Engineering Risk Assessment

Use when: A new initiative or component carries engineering risk that needs surfacing before commitment.

Outcome: A structured engineering risk assessment distinct from delivery/schedule risk, covering technical unknowns.

Inputs: Initiative description, [SYSTEMS] involved, team's prior experience with the approach

PROMPT

```
CONTEXT: Assessing engineering risk for <initiative/component> on [PROJECT] involving [SYSTEMS].
OBJECTIVE: Surface technical unknowns, integration risks and skill-gap risks before commitment,
distinct from delivery scheduling risk.
INPUTS: Initiative/technical approach description, systems involved, team's prior experience with
similar work.
CONSTRAINTS: Rate a risk HIGH only where the team lacks prior experience AND the failure mode has
significant consequence — do not rate familiar, low-consequence work as high risk by default
caution.
DECISION RULES: Flag SPIKE NEEDED if a technical unknown cannot be resolved by design review
alone and requires a timeboxed investigation before committing an estimate.
OUTPUT CONTRACT: Table [Risk | Category (Unknown Tech/Integration/Skill Gap/Scale) | Consequence
if Realized | Spike Needed? (Y/N) | Recommended Mitigation].
VALIDATION: Confirm HIGH ratings show both low familiarity and high consequence, not either
alone.
ESCALATION: Spike-needed items route into the next iteration's plan before full estimation.
```

Output: Engineering risk assessment with spike recommendations where warranted.

Validate: HIGH ratings checked against both familiarity and consequence factors.

Next: Spike-needed items scheduled before TL-11 Engineering Estimation.

LEVEL L1 HUMAN GATE No

TL-05 API Design Review

Use when: An API design needs review for consistency, versioning and consumer impact before publication.

Outcome: A structured API review covering contract design, versioning strategy and breaking-change risk.

Inputs: API spec (endpoints/schema), existing API standards, known consumers

PROMPT

```
CONTEXT: Reviewing the API design for <API name> on [PROJECT] against existing standards and
known consumers.
OBJECTIVE: Identify contract design issues, versioning gaps and breaking-change risk for existing
consumers.
INPUTS: API specification, organizational API standards if documented, list of known/expected
consumers.
CONSTRAINTS: Flag deviations from documented standards explicitly; do not assume an undocumented
"common sense" standard.
DECISION RULES: Classify a change as BREAKING if it removes/renames a field, changes a type, or
alters required/optional status for existing consumers; otherwise NON-BREAKING.
OUTPUT CONTRACT: Table [Element | Issue | Standard Reference (or none) | Breaking? (Y/N) |
Recommendation] + versioning strategy recommendation.
VALIDATION: Every BREAKING classification checked against the exact consumer-facing contract
change, not internal implementation change.
ESCALATION: Breaking changes to consumers outside the team route to a consumer-
notification/migration plan before release.
```

Output: API design review with explicit breaking-change classification and versioning guidance.

Validate: Breaking-change classifications checked against actual contract impact.

Next: Breaking changes trigger a consumer migration/communication plan.

LEVEL L1 HUMAN GATE Conditional

TL-06 Integration Design Analysis

Use when: Two or more systems need to integrate and you need the design analyzed for failure modes and coupling risk.

Outcome: An integration design analysis covering coupling, failure handling and data consistency risk.

Inputs: Integration design/sequence diagrams, [SYSTEMS] involved, [CONSTRAINTS] (SLAs, data volume)

PROMPT

```
CONTEXT: Analyzing the integration design between [SYSTEMS] for [PROJECT].
OBJECTIVE: Identify coupling risk, failure handling gaps and data consistency risk in the
proposed integration.
INPUTS: Integration design/sequence diagrams, SLA/throughput expectations, data consistency
requirements.
CONSTRAINTS: Evaluate synchronous vs asynchronous choice against stated SLA and volume needs — do
not assume synchronous is always simpler/better.
DECISION RULES: Flag TIGHT COUPLING RISK if the design has no fallback/retry/circuit-breaker for
a synchronous dependency. Flag CONSISTENCY RISK if eventual consistency is used where the
requirement implies strong consistency.
OUTPUT CONTRACT: Table [Integration Point | Pattern Used | Failure Handling Present? | Risk |
Recommendation].
VALIDATION: Confirm each risk flag traces to the specific stated SLA/consistency requirement it
conflicts with.
ESCALATION: Consistency risks affecting financial/customer data route to SA/Data Architect
review.
```

Output: Integration risk analysis covering coupling, failure handling and consistency.

Validate: Risk flags traced to the specific conflicting SLA/consistency requirement.

Next: Consistency-critical risks route to DA-01/SA-04 for architecture-level review.

LEVEL L1 HUMAN GATE No

TL-07 Scalability Assessment

Use when: A component's ability to handle projected growth needs objective assessment before it becomes a production incident.

Outcome: A scalability assessment identifying the specific bottleneck and headroom, not a general 'should be fine.'

Inputs: Current architecture/component design, current load data, projected growth ([THRESHOLD])

PROMPT

```
CONTEXT: Assessing scalability of <component/service> on [PROJECT] against projected growth to
[THRESHOLD].
OBJECTIVE: Identify the specific bottleneck resource (CPU, memory, DB connections, I/O, a
downstream dependency) and current headroom.
INPUTS: Current architecture, current load/performance data, projected growth target.
CONSTRAINTS: Base headroom estimates on actual measured resource utilization at current load — do
not estimate from architecture diagrams alone.
DECISION RULES: Flag AT RISK if projected load exceeds current measured capacity of any single
resource by [THRESHOLD] with no scaling mechanism (horizontal/vertical) in place.
OUTPUT CONTRACT: Table [Resource/Component | Current Utilization at Current Load | Projected
Utilization at [THRESHOLD] | Scaling Mechanism Present? | Flag] + primary bottleneck statement.
VALIDATION: Confirm the projected-utilization figures are calculated (not guessed) from current
measured data scaled linearly or per stated model.
ESCALATION: AT RISK components with no scaling mechanism and near-term growth route to
architecture review before the growth event.
```

Output: Bottleneck-specific scalability assessment with calculated headroom.

Validate: Projected utilization figures shown as a calculation from current measured data.

Next: AT RISK items route to CA-07/SA-06 for scaling architecture design.

LEVEL L2 HUMAN GATE No

TL-08 Performance Bottleneck Analysis

Use when: A system is exhibiting performance issues and you need root cause, not symptom treatment.

Outcome: A root-cause performance diagnosis with the specific slow component/query/path identified.

Inputs: Performance data (traces, logs, profiling output), the reported symptom, [SUCCESS_CRITERIA] (target latency/throughput)

PROMPT

```
CONTEXT: Diagnosing a performance issue on [SYSTEMS] for [PROJECT]: <describe symptom> against
target [SUCCESS_CRITERIA].
OBJECTIVE: Identify the specific root-cause bottleneck (not just the symptomatic layer) using
available performance data.
INPUTS: Traces/profiling/logs, the reported symptom, target performance criteria.
CONSTRAINTS: Distinguish the bottleneck (root cause) from where the symptom is observed (e.g.,
"API times out" is a symptom; "unindexed query on table X" may be the cause) — do not conflate
the two.
DECISION RULES: Root cause = the component/step consuming the largest share of total latency per
the trace/profile data.
OUTPUT CONTRACT: Sections — Symptom | Trace/Profile Findings (with % of latency per component) |
Root Cause | Recommended Fix | Expected Improvement.
VALIDATION: Confirm the root cause is the highest-latency-share component per the data shown, not
an assumption.
ESCALATION: If profiling data is insufficient to isolate root cause, flag DATA GAP and specify
what additional instrumentation is needed.
```

Output: Root-cause performance diagnosis with latency-share evidence and a recommended fix.

Validate: Root cause confirmed as the highest latency-share component from the data.

Next: Fix verified against target [SUCCESS_CRITERIA] post-implementation.

LEVEL L1 HUMAN GATE No

TL-09 Reliability Assessment

Use when: You need to assess a system's reliability posture — failure modes, redundancy, recovery — before a criticality upgrade.

Outcome: A reliability assessment naming specific single points of failure and recovery time gaps.

Inputs: Architecture diagram, [SYSTEMS], target [SUCCESS_CRITERIA] (uptime/RTO/RPO)

PROMPT

```
CONTEXT: Assessing reliability of [SYSTEMS] on [PROJECT] against target uptime/RTO/RPO in
[SUCCESS_CRITERIA].
OBJECTIVE: Identify specific single points of failure and gaps versus the target recovery
objectives.
INPUTS: Architecture diagram/component list, redundancy configuration, current backup/recovery
procedures.
CONSTRAINTS: A component is a SPOF only if its failure has no automatic failover/redundancy — do
not flag components with confirmed redundancy.
DECISION RULES: Flag RTO/RPO GAP if the documented recovery procedure's tested/measured time
exceeds the target in [SUCCESS_CRITERIA].
OUTPUT CONTRACT: Table [Component | Redundancy Present? | SPOF? | Recovery Time
(measured/estimated) | Gap vs Target | Recommendation].
VALIDATION: SPOF flags confirmed against actual redundancy configuration, not architecture
diagram intent.
ESCALATION: Untested recovery procedures for critical components route to a DR test exercise
before sign-off.
```

Output: SPOF and RTO/RPO gap assessment against target reliability criteria.

Validate: SPOF flags checked against actual, not intended, redundancy configuration.

Next: Gaps feed CA-08 DR Strategy Design or IA-05 HA/DR Design Review.

LEVEL L2 HUMAN GATE No

TL-10 CI/CD Pipeline Assessment

Use when: Build/deploy friction or failures are slowing the team and you need a structured pipeline health check.

Outcome: A CI/CD health assessment identifying the specific stage causing friction, with fix recommendations.

Inputs: Pipeline configuration, build/deploy history data, failure logs

PROMPT

```
CONTEXT: Assessing CI/CD pipeline health for [PROJECT] based on recent build/deploy history.
OBJECTIVE: Identify the specific pipeline stage(s) causing failures or delay, with root cause and
fix.
INPUTS: Pipeline configuration, build success/failure history, average duration per stage.
CONSTRAINTS: Identify the highest-friction stage using actual failure-rate and duration data, not
general impression of "the tests are flaky."
DECISION RULES: Flag a stage UNSTABLE if failure rate >15% over the analysis period; flag SLOW if
its duration exceeds 40% of total pipeline time.
OUTPUT CONTRACT: Table [Stage | Failure Rate | Avg Duration | % of Total Pipeline Time | Flag |
Root Cause | Recommendation].
VALIDATION: Failure rate and duration percentages recalculated from the raw history data
provided.
ESCALATION: Persistent instability blocking releases routes to OPS-01 CI/CD Pipeline Design
Review for structural redesign.
```

Output: Data-driven CI/CD stage diagnostic with fix recommendations.

Validate: Failure-rate and duration percentages recalculated from raw history data.

Next: Structural issues escalate to OPS-01 for pipeline redesign.

LEVEL L1 HUMAN GATE No

TL-11 Engineering Estimation

Use when: A scoped piece of work needs a defensible estimate with stated assumptions and confidence.

Outcome: An estimate with explicit assumptions, confidence range and the specific unknowns driving uncertainty.

Inputs: Scoped work description, [SYSTEMS] context, team's historical estimation accuracy if known

PROMPT

```
CONTEXT: Estimating effort for <scoped work item> on [PROJECT].
OBJECTIVE: Produce an estimate range with explicit assumptions and the specific unknowns driving
the range's width.
INPUTS: Scoped work description, related design/technical review outputs, historical estimation
accuracy if tracked.
CONSTRAINTS: State every assumption the estimate depends on explicitly — do not bury assumptions
inside a single point number.
DECISION RULES: Provide a range (low/likely/high), not a single point estimate, unless historical
accuracy data supports high confidence in a point figure.
OUTPUT CONTRACT: Sections — Scope Assumed | Estimate Range (low/likely/high) | Key Unknowns
Driving the Range | Confidence Level (with basis) | Dependencies Assumed Resolved.
VALIDATION: Confirm the range width is justified by the stated unknowns, not an arbitrary padding
percentage.
ESCALATION: Low-confidence estimates on committed-date work route to TL-04 Engineering Risk
Assessment for spike planning first.
```

Output: Range-based estimate with explicit assumptions and confidence basis.

Validate: Range width justified by named unknowns, not arbitrary padding.

Next: Low-confidence items route to TL-04 for a spike before committing dates.

LEVEL L1 HUMAN GATE No

TL-12 Build-vs-Buy Analysis

Use when: A capability need could be built in-house or bought/adopted, and the decision needs rigor.

Outcome: A structured build-vs-buy analysis comparing total cost, risk and strategic fit, not just sticker price.

Inputs: Capability requirement, candidate buy/adopt options, [CONSTRAINTS] (budget, timeline, team skill)

PROMPT

```
CONTEXT: Evaluating build-vs-buy for <capability> on [PROJECT] under [CONSTRAINTS].
OBJECTIVE: Compare build vs buy/adopt options on total cost of ownership, time-to-value, risk and
strategic fit.
INPUTS: Capability requirement, candidate vendor/OSS options if buying, team skill/capacity if
building.
CONSTRAINTS: Total cost must include ongoing maintenance/support, not just initial build/license
cost.
DECISION RULES: Favor BUY if the capability is non-differentiating and a mature option meets
[SUCCESS_CRITERIA] within [CONSTRAINTS]; favor BUILD if it is a core differentiator or no option
meets a hard requirement.
OUTPUT CONTRACT: Comparison table [Option | Initial Cost | Ongoing Cost | Time to Value | Risk |
Strategic Fit] + recommendation with rationale.
VALIDATION: Confirm ongoing cost figures are included for every option, not just initial cost.
ESCALATION: Decisions with material budget or vendor lock-in implications route to Enterprise
Architect / procurement for sign-off.
```

Output: Total-cost-of-ownership build-vs-buy comparison with a clear recommendation.

Validate: Ongoing/maintenance costs confirmed present for every compared option.

Next: Material decisions route to EA-03/procurement for final sign-off.

LEVEL L1 HUMAN GATE Yes

Solution Architect (SA)

Architecture options, trade-offs and decisions grounded in explicit NFRs and evidence, not preference.

SA-01 Architecture Options Analysis

Use when: A solution needs multiple credible architecture options compared before committing.

Outcome: 2-3 genuinely distinct architecture options with trade-offs, not one option and two strawmen.

Inputs: [OBJECTIVE], NFRs/[SUCCESS_CRITERIA], [CONSTRAINTS], [SYSTEMS] context

PROMPT

```
CONTEXT: Developing architecture options for [OBJECTIVE] on [PROJECT], within [CONSTRAINTS].
OBJECTIVE: Present 2–3 genuinely distinct, viable architecture options with explicit trade-offs.
INPUTS: Requirements/NFRs, constraints (budget, timeline, existing landscape), [SYSTEMS] to
integrate with.
CONSTRAINTS: Every option must be independently viable — no option included purely as a strawman
to make another look better.
DECISION RULES: Score each option against the same NFR/criteria set for a fair comparison; do not
use different evaluation lenses per option.
OUTPUT CONTRACT: Table [Option | Description | NFR Fit (per criterion) | Cost/Effort | Risk |
Strategic Fit] + recommendation with explicit rationale and what would change the recommendation.
VALIDATION: Confirm all options were scored against the identical criteria set.
ESCALATION: Options with materially different cost/risk profiles route to sponsor/governance for
a funded decision.
```

Output: Scored, comparable architecture options with a clear recommendation.

Validate: All options confirmed scored against the identical criteria set.

Next: Feed into SA-12 ADR to record the decision.

LEVEL L2 HUMAN GATE Yes

SA-02 Architecture Trade-off Matrix

Use when: Two or more NFRs are in tension (e.g., cost vs resilience) and the trade-off needs to be explicit.

Outcome: A trade-off matrix quantifying the tension between competing quality attributes.

Inputs: Competing NFRs, [CONSTRAINTS], [RISK_TOLERANCE]

PROMPT

```
CONTEXT: Resolving a trade-off between <NFR A> and <NFR B> for [PROJECT] architecture.
OBJECTIVE: Quantify how improving one quality attribute affects the other, to support an informed
trade-off decision.
INPUTS: The competing NFRs with their targets, relevant constraints, risk tolerance.
CONSTRAINTS: State the trade-off mechanism explicitly (e.g., "adding redundancy improves
availability but increases cost and data-consistency complexity") — don't just assert a tension
exists.
DECISION RULES: Recommend the point on the trade-off curve that satisfies the minimum acceptable
threshold for both NFRs per [CONSTRAINTS]/[RISK_TOLERANCE], not the theoretical optimum for
either alone.
OUTPUT CONTRACT: Trade-off table [Design Point | NFR A Outcome | NFR B Outcome | Cost | Risk] +
recommended point with rationale.
VALIDATION: Confirm the recommended point meets both stated minimum thresholds, not just one.
ESCALATION: If no design point meets both minimums within [CONSTRAINTS], escalate the constraint
conflict to sponsor.
```

Output: Quantified trade-off matrix with a recommended design point.

Validate: Recommended point checked against both stated minimum NFR thresholds.

Next: Document the chosen point via SA-12 ADR.

LEVEL L1 HUMAN GATE Yes

SA-03 NFR Definition & Validation

Use when: NFRs from BA-07 need architectural feasibility validation before being locked in.

Outcome: A feasibility-validated NFR set, with any infeasible target flagged and an alternative proposed.

Inputs: Draft NFRs (BA-07), current/target architecture, [CONSTRAINTS]

PROMPT

```
CONTEXT: Validating feasibility of draft NFRs for [SYSTEMS] on [PROJECT] within [CONSTRAINTS].
OBJECTIVE: Confirm each NFR target is architecturally achievable, or propose a feasible
alternative with rationale.
INPUTS: Draft NFR set, current architecture capability, constraints (budget, timeline,
technology).
CONSTRAINTS: Base feasibility judgment on the actual current architecture's demonstrated
capability or documented capability of the proposed target architecture — not aspiration.
DECISION RULES: Mark FEASIBLE if achievable within [CONSTRAINTS] using known patterns; mark AT
RISK if achievable only with significant investment; mark INFEASIBLE if no known approach meets
it within constraints.
OUTPUT CONTRACT: Table [NFR | Target | Feasibility | Basis | Alternative Target (if infeasible) |
Investment Needed (if at risk)].
VALIDATION: Every INFEASIBLE or AT RISK verdict cites the specific constraint or capability
limitation driving it.
ESCALATION: INFEASIBLE NFRs route back to BA/PO and sponsor for a target renegotiation before
requirements sign-off.
```

Output: Feasibility-validated NFR set with alternatives for infeasible targets.

Validate: Infeasible/at-risk verdicts cite the specific limiting constraint.

Next: Infeasible items route back to BA-07/PO-02 for renegotiation.

LEVEL L2 HUMAN GATE Yes

SA-04 Integration Architecture Design

Use when: Designing the integration approach across multiple systems for a new capability.

Outcome: An integration architecture with pattern choice explicitly justified against data volume, latency and consistency needs.

Inputs: [SYSTEMS] to integrate, data volume/latency requirements, [CONSTRAINTS]

PROMPT

```
CONTEXT: Designing integration architecture across [SYSTEMS] for [PROJECT].
OBJECTIVE: Select and justify an integration pattern (sync API, async messaging, event-driven,
batch/ETL) against actual requirements.
INPUTS: Systems involved, data volume/frequency, latency tolerance, consistency requirements.
CONSTRAINTS: Justify the pattern choice against the stated volume/latency/consistency needs — do
not default to "REST API" without checking fit.
DECISION RULES: Use synchronous API only where real-time response is required and volume is
manageable; use async/event-driven for high volume, decoupling needs, or eventual-consistency-
tolerant flows; use batch for large volume with no real-time need.
OUTPUT CONTRACT: Sections — Integration Requirements Summary | Pattern Selected with Rationale |
Data Flow Diagram (described) | Failure Handling Approach | Rejected Alternatives with Reason.
VALIDATION: Confirm the selected pattern's characteristics (latency, consistency model) match the
stated requirement, not just convention.
ESCALATION: High-volume financial/customer-data integrations route to Data Architect/Security
Architect for joint review.
```

Output: Justified integration architecture with explicit pattern rationale.

Validate: Selected pattern's characteristics checked against actual stated requirements.

Next: Joint review with DA/SEC for sensitive data integrations.

LEVEL L2 HUMAN GATE No

SA-05 API Strategy Design

Use when: Defining or evolving an organization/product's API strategy beyond a single API's design.

Outcome: An API strategy covering versioning, consumer management and governance model, not just endpoint design.

Inputs: Current API landscape, consumer base/types, [CONSTRAINTS]

PROMPT

```
CONTEXT: Defining API strategy for [PRODUCT]/[SYSTEMS] on [PROJECT].
OBJECTIVE: Define versioning policy, consumer management approach and governance model for APIs
at a program/product level.
INPUTS: Current API inventory, internal vs external consumer mix, known pain points (breaking
changes, inconsistent standards).
CONSTRAINTS: Versioning policy must specify a deprecation timeline and communication method —
"we'll version when needed" is not acceptable output.
DECISION RULES: Recommend a versioning scheme (e.g., URI versioning, header versioning) based on
consumer type mix — external/public APIs typically need more conservative, longer-lived
versioning than internal-only APIs.
OUTPUT CONTRACT: Sections — Current State Summary | Versioning Policy (scheme + deprecation
timeline) | Governance Model (who approves new/breaking changes) | Consumer Communication Plan.
VALIDATION: Confirm the versioning policy specifies both a scheme and a deprecation timeframe,
not just a scheme name.
ESCALATION: Governance model changes affecting external partners route to customer/partner
engagement for review.
```

Output: API strategy covering versioning, governance and consumer communication.

Validate: Versioning policy confirmed to include both scheme and deprecation timeframe.

Next: Apply per-API via TL-05 API Design Review.

LEVEL L2 HUMAN GATE Yes

SA-06 Scalability Architecture Review

Use when: You need an architecture-level (not single-component) view of scalability across a solution.

Outcome: A solution-wide scalability review identifying the architectural constraint, not just a hot component.

Inputs: Solution architecture, projected load ([THRESHOLD]), current scaling mechanisms per tier

PROMPT

```
CONTEXT: Reviewing solution-wide scalability for [PROJECT] against projected load [THRESHOLD].
OBJECTIVE: Identify the architectural tier (presentation, application, data, integration) that
constrains overall scale, and the mechanism to relieve it.
INPUTS: Solution architecture by tier, current scaling approach (horizontal/vertical/none) per
tier, projected load target.
CONSTRAINTS: Evaluate the data tier's scaling limits explicitly — this is the most common
architecture-level constraint and must not be assumed "handled" without evidence.
DECISION RULES: The architectural bottleneck is the tier with the lowest scaling ceiling relative
to [THRESHOLD]; flag it as the PRIMARY CONSTRAINT.
OUTPUT CONTRACT: Table [Tier | Current Scaling Mechanism | Scaling Ceiling (with basis) | Meets
[THRESHOLD]? | Flag] + primary constraint statement and remediation options.
VALIDATION: Confirm the data tier was explicitly evaluated, not assumed scalable by default.
ESCALATION: PRIMARY CONSTRAINT requiring re-architecture routes to sponsor for investment
decision (see SA-01 Options Analysis).
```

Output: Tier-by-tier scalability review with a named primary architectural constraint.

Validate: Data tier explicitly evaluated, not assumed scalable by default.

Next: Primary constraint feeds SA-01 Architecture Options Analysis.

LEVEL L2 HUMAN GATE No

SA-07 Reliability Architecture Review

Use when: Assessing solution-wide reliability posture across multiple components/tiers.

Outcome: A reliability review with a calculated composite availability estimate, not just per-component checks.

Inputs: Architecture diagram with component dependencies, per-component availability/SLA, target [SUCCESS_CRITERIA]

PROMPT

```
CONTEXT: Reviewing solution-wide reliability for [PROJECT] against target availability in
[SUCCESS_CRITERIA].
OBJECTIVE: Calculate composite availability across the dependency chain and identify the
component(s) driving it down.
INPUTS: Architecture with dependency chain, per-component SLA/availability figures.
CONSTRAINTS: Composite availability for a serial dependency chain is the product of individual
component availabilities — show this calculation, do not assume the weakest link alone determines
it without showing the math for chains.
DECISION RULES: Flag GAP if calculated composite availability is below target in
[SUCCESS_CRITERIA]; identify which component contributes most to the shortfall.
OUTPUT CONTRACT: Dependency chain table [Component | Availability | Redundant Path? (removes it
from the serial calc if yes)] + calculated composite availability + gap statement + top
remediation lever.
VALIDATION: Composite availability calculation shown step by step, reproducible from the
component figures.
ESCALATION: Availability gaps affecting contractual SLAs route to sponsor/legal for risk
acceptance or investment decision.
```

Output: Calculated composite availability with the primary shortfall driver identified.

Validate: Composite calculation shown step by step and independently reproducible.

Next: Remediation levers feed SA-01 Options Analysis or CA-07 Resilience Design.

LEVEL L2 HUMAN GATE No

SA-08 Security Architecture Review (Solution-Level)

Use when: A solution design needs a high-level security posture review before detailed security architecture work.

Outcome: A solution-level security review identifying architectural exposure, routed to Security Architect for depth.

Inputs: Solution architecture, data classification, [SYSTEMS] trust boundaries

PROMPT

```
CONTEXT: Reviewing solution-level security posture for [PROJECT] handling data classified as
<classification>.
OBJECTIVE: Identify architectural security exposure (trust boundary gaps, unencrypted data paths,
missing auth checkpoints) at the solution design level.
INPUTS: Solution architecture with trust boundaries marked, data classification, known
integration points.
CONSTRAINTS: This is a solution-architecture-level pass, not a full threat model — flag items for
SEC-01 Threat Modeling rather than attempting exhaustive analysis here.
DECISION RULES: Flag EXPOSURE if a trust boundary crossing lacks an explicit
authentication/authorization checkpoint, or if classified data traverses a path without stated
encryption.
OUTPUT CONTRACT: Table [Trust Boundary/Data Path | Control Present? | Exposure Flag |
Recommendation] + explicit hand-off note to Security Architect for deep-dive items.
VALIDATION: Every EXPOSURE flag names the specific missing control (encryption, authN, authZ),
not a general "review security."
ESCALATION: All EXPOSURE flags route to SEC-01 Threat Modeling and SEC-02 Security Architecture
Review before build sign-off.
```

Output: Solution-level security exposure list handed off for deep security review.

Validate: Exposure flags name the specific missing control, not general concern.

Next: Hand off to SEC-01 Threat Modeling for depth.

LEVEL L2 HUMAN GATE Yes

SA-09 Technology Selection Matrix

Use when: Choosing between technology/platform options and need a defensible, weighted comparison.

Outcome: A weighted technology selection matrix reflecting actual organizational priorities, not generic best-practice weighting.

Inputs: Candidate technologies, evaluation criteria and their [BUSINESS_OUTCOME]-driven weights, [CONSTRAINTS]

PROMPT

```
CONTEXT: Selecting a technology/platform for <capability> on [PROJECT] to serve
[BUSINESS_OUTCOME].
OBJECTIVE: Score candidate technologies against weighted criteria reflecting actual
organizational priorities.
INPUTS: Candidate list, evaluation criteria (e.g., cost, skill availability, ecosystem maturity,
vendor lock-in, performance fit), stated weight/importance of each.
CONSTRAINTS: Weights must be explicit and justified against [BUSINESS_OUTCOME]/[CONSTRAINTS] — do
not use unweighted or equally-weighted criteria by default without confirming that reflects
priorities.
DECISION RULES: Weighted score = Σ(criterion score × weight); highest weighted score wins unless
a criterion is marked as a hard gate (disqualifying if failed).
OUTPUT CONTRACT: Matrix [Criterion | Weight | Option A Score | Option B Score | Option C Score] +
weighted totals + hard-gate check + recommendation.
VALIDATION: Recompute the winning option's weighted total manually to confirm arithmetic.
ESCALATION: Selections with material vendor lock-in or long-term cost implications route to
Enterprise Architect/procurement for sign-off.
```

Output: Weighted technology selection matrix with a defensible recommendation.

Validate: Winning weighted total manually recomputed to confirm accuracy.

Next: Material lock-in decisions route to EA/procurement sign-off.

LEVEL L1 HUMAN GATE Yes

SA-10 Feasibility Assessment

Use when: Before committing to an initiative, you need a fast, honest feasibility read across technical, resource and timeline dimensions.

Outcome: A feasibility verdict per dimension with the specific blocking factor named where infeasible.

Inputs: Initiative description, [CONSTRAINTS], current [SYSTEMS]/team capability

PROMPT

```
CONTEXT: Assessing feasibility of <initiative> for [PROJECT] within [CONSTRAINTS].
OBJECTIVE: Give an honest feasibility verdict across technical, resourcing and timeline
dimensions, naming specific blockers where infeasible.
INPUTS: Initiative description/requirements, known constraints, current system/team capability.
CONSTRAINTS: A FEASIBLE verdict requires no known blocking factor across all three dimensions —
do not rate feasible with an unaddressed blocker "to be solved later."
DECISION RULES: INFEASIBLE if a hard technical, regulatory or resourcing blocker exists with no
workaround within [CONSTRAINTS]; CONDITIONAL if feasible only with a specific named change
(budget, timeline, scope); FEASIBLE otherwise.
OUTPUT CONTRACT: Table [Dimension | Verdict | Specific Blocker (if any) | Condition Needed (if
Conditional)] + overall verdict.
VALIDATION: Every INFEASIBLE/CONDITIONAL verdict names the specific blocking factor, not a
general "may be difficult."
ESCALATION: INFEASIBLE overall verdicts route to sponsor before any further planning investment.
```

Output: Dimension-by-dimension feasibility verdict with named blockers.

Validate: Infeasible/conditional verdicts each name a specific blocking factor.

Next: Feasible/conditional items proceed to SA-01 Architecture Options.

LEVEL L1 HUMAN GATE Yes

SA-11 POC Evaluation

Use when: A proof-of-concept has run and its results need objective evaluation against the original success criteria.

Outcome: An objective POC evaluation against pre-defined success criteria, resisting sunk-cost bias.

Inputs: Original POC success criteria, actual POC results/data, [CONSTRAINTS]

PROMPT

```
CONTEXT: Evaluating the POC for <initiative/technology> on [PROJECT] against its original success criteria.
OBJECTIVE: Determine objectively whether the POC met, partially met, or failed its pre-defined success criteria.
INPUTS: Original success criteria (defined before the POC ran), actual results/data from the POC.
CONSTRAINTS: Evaluate strictly against the criteria as originally defined — do not retroactively adjust criteria to fit the result (sunk-cost/confirmation bias check).
DECISION RULES: MET if measured result meets or exceeds each criterion's threshold; PARTIALLY MET if some but not all criteria met; NOT MET if primary criteria failed.
OUTPUT CONTRACT: Table [Success Criterion (original) | Target | Actual Result | Met? (Y/N)] + overall verdict + recommendation (Proceed/Proceed with changes/Do not proceed).
VALIDATION: Confirm the criteria list used matches the originally defined set, with no post-hoc additions or removals.
ESCALATION: NOT MET verdicts on initiatives with sunk investment route to sponsor for an explicit "stop" decision rather than silent continuation.
```

Output: Objective POC verdict against pre-defined criteria with a clear recommendation.

Validate: Criteria list confirmed unchanged from the original pre-POC definition.

Next: Proceed decisions feed SA-01 Options Analysis for full architecture design.

LEVEL L1 HUMAN GATE Yes

SA-12 Architecture Decision Record (ADR) Generator

Use when: A significant architecture decision has been made and needs to be recorded for future reference.

Outcome: A concise, standard-format ADR capturing decision, context, and consequences for future teams.

Inputs: The decision, context/options considered, [CONSTRAINTS] that shaped it

PROMPT

```
CONTEXT: Recording an architecture decision for [PROJECT]: <state the decision>.
OBJECTIVE: Produce a standard-format ADR that lets a future team understand why this decision was made without re-litigating it.
INPUTS: The decision, alternatives considered (e.g., from SA-01), constraints that drove the choice.
CONSTRAINTS: Include the alternatives that were rejected and why — an ADR that only states the chosen option loses its future value.
DECISION RULES: Status must be one of Proposed/Accepted/Deprecated/Superseded — never left blank.
OUTPUT CONTRACT: Standard sections — Title | Status | Context | Decision | Alternatives Considered (with rejection reason) | Consequences (positive and negative) | Date.
VALIDATION: Confirm at least one rejected alternative is documented with its specific rejection reason.
ESCALATION: N/A — ADRs are a record, not a decision gate; but publication should be reviewed by the architecture review board if one exists.
```

Output: Standard-format ADR with alternatives, rationale and consequences.

Validate: At least one rejected alternative documented with a specific reason.

Next: Store in the architecture decision log; link from SA-01/SA-02 outputs.

LEVEL L1 HUMAN GATE No

Cloud Architect (CA)

Cloud architecture, migration, cost and resilience decisions grounded in Well-Architected discipline.

CA-01 Cloud Architecture Design

Use when: Designing a cloud-native or cloud-target architecture for a new or migrating workload.

Outcome: A cloud architecture design with service choices explicitly justified against NFRs and cost.

Inputs: Workload requirements/NFRs, [CONSTRAINTS] (cloud provider, budget), [SYSTEMS] context

PROMPT

```
CONTEXT: Designing cloud architecture for <workload> on [PROJECT] within [CONSTRAINTS].
OBJECTIVE: Select and justify cloud services (compute, storage, networking, data) against stated
NFRs and cost targets.
INPUTS: Workload requirements/NFRs, provider constraints, integration context.
CONSTRAINTS: Justify managed-service vs self-managed choices explicitly against operational
overhead and cost — do not default to the most feature-rich managed option without cost
justification.
DECISION RULES: Prefer managed services where operational overhead reduction outweighs the cost
premium and no NFR is compromised; prefer self-managed only where a specific NFR or cost
constraint requires it.
OUTPUT CONTRACT: Sections — Workload Summary | Service Selection (per tier, with rationale) |
Estimated Cost Profile | NFR Fit Check | Rejected Alternatives.
VALIDATION: Confirm every service choice's rationale references a specific NFR or cost figure,
not general preference.
ESCALATION: Cost estimates exceeding budget constraints route to sponsor for trade-off discussion
(CA-06).
```

Output: Justified cloud architecture design with cost profile and NFR fit check.

Validate: Each service choice rationale references a specific NFR or cost figure.

Next: Cost overruns route to CA-06 Cost Optimization Analysis.

LEVEL L2 HUMAN GATE No

CA-02 Migration Strategy & Wave Plan

Use when: Migrating multiple workloads to cloud and need a sequenced, risk-aware wave plan.

Outcome: A wave-based migration plan sequenced by risk, dependency and business value, not alphabetically.

Inputs: Workload inventory, dependency map, [BUSINESS_OUTCOME], [CONSTRAINTS] (timeline/budget)

PROMPT

```
CONTEXT: Planning cloud migration waves for <workload portfolio> under [PROGRAM], target
[BUSINESS_OUTCOME].
OBJECTIVE: Sequence workloads into migration waves based on dependency, risk and business value —
not convenience.
INPUTS: Workload inventory with dependencies, criticality/business value, complexity indicators
(legacy tech, custom integrations).
CONSTRAINTS: A workload cannot be scheduled before its upstream dependencies unless a documented
compatibility bridge exists.
DECISION RULES: Wave 1 = low-complexity, low-risk, dependency-independent workloads (build
confidence/pattern); later waves = higher complexity/risk, sequenced by dependency order.
OUTPUT CONTRACT: Table [Workload | Wave | Migration Pattern (rehost/replatform/refactor) |
Dependency Constraint | Risk Level | Rationale].
VALIDATION: Confirm no workload is scheduled ahead of an unresolved upstream dependency.
ESCALATION: High-risk workloads with no clear migration pattern route to CA-01/SA-01 for design
work before wave scheduling.
```

Output: Risk- and dependency-sequenced migration wave plan.

Validate: Dependency ordering confirmed with no workload ahead of unresolved upstream needs.

Next: High-risk workloads get dedicated design work (CA-01) before their wave.

LEVEL L2 HUMAN GATE Yes

CA-03 Modernization Assessment

Use when: Deciding how much to modernize a workload during/after cloud migration (the 6 R's decision).

Outcome: A per-workload modernization recommendation (rehost/replatform/refactor/etc.) with explicit rationale.

Inputs: Workload technical profile, [BUSINESS_OUTCOME], [CONSTRAINTS] (timeline, budget, risk appetite)

PROMPT

```
CONTEXT: Assessing modernization approach for <workload> as part of [PROGRAM] cloud migration.
OBJECTIVE: Recommend one of Rehost/Replatform/Refactor/Repurchase/Retire/Retain, justified
against business value and technical fit.
INPUTS: Workload's technical debt profile, business criticality/value, remaining useful life
expectation, budget/timeline constraints.
CONSTRAINTS: Recommend Refactor only where business value justifies the higher investment — do
not default to "always refactor" or "always rehost."
DECISION RULES: RETIRE if usage/value data shows the workload is obsolete; RETAIN if
regulatory/technical constraints prevent migration; REHOST for low-value/urgent-timeline cases;
REFACTOR for high-value, high-debt cases with runway to invest.
OUTPUT CONTRACT: Table [Workload | Business Value | Technical Debt Level | Recommendation |
Rationale | Estimated Effort Tier].
VALIDATION: Confirm each recommendation follows the stated decision rule against the value/debt
inputs, not a default pattern.
ESCALATION: RETIRE recommendations route to business owner for confirmation before decommission
planning.
```

Output: Justified per-workload modernization recommendation across the 6 R's.

Validate: Each recommendation checked against the value/debt decision rule.

Next: Retire candidates route to business owner sign-off before CA-02 wave planning excludes them.

LEVEL L2 HUMAN GATE Yes

CA-04 Landing Zone Design

Use when: Standing up or restructuring the foundational cloud environment (accounts, networking, guardrails).

Outcome: A landing zone design covering account structure, networking and guardrails, matched to governance needs.

Inputs: [CUSTOMER]/organization structure, compliance requirements, expected workload types

PROMPT

```
CONTEXT: Designing the cloud landing zone for [CUSTOMER]/[PROGRAM].
OBJECTIVE: Define account/subscription structure, network topology and policy guardrails matched
to governance and compliance needs.
INPUTS: Organizational structure (business units, environments), compliance/regulatory
requirements, expected workload diversity.
CONSTRAINTS: Account/subscription segmentation must map to actual governance boundaries (billing,
compliance scope, access control) — not an arbitrary team-based split.
DECISION RULES: Separate accounts/subscriptions by environment (prod/non-prod) at minimum, and by
compliance scope where regulatory isolation is required.
OUTPUT CONTRACT: Sections — Account/Subscription Structure (with rationale) | Network Topology
Summary | Guardrail Policies (by category: security, cost, tagging) | Compliance Mapping.
VALIDATION: Confirm every guardrail policy maps to a specific stated compliance or governance
requirement.
ESCALATION: Compliance scope decisions route to Security Architect and legal/compliance for sign-
off.
```

Output: Landing zone design with account structure, guardrails and compliance mapping.

Validate: Guardrail policies confirmed to map to specific compliance/governance requirements.

Next: Security review via CA-09; compliance sign-off before build.

LEVEL L2 HUMAN GATE Yes

CA-05 Well-Architected Review

Use when: Periodic or pre-launch review of a workload against the cloud provider's Well-Architected pillars.

Outcome: A pillar-by-pillar review with specific, prioritized remediation items — not a generic checklist pass.

Inputs: Workload architecture, current configuration data, [THRESHOLD] for pillar risk acceptance

PROMPT

```
CONTEXT: Conducting a Well-Architected review of <workload> on [PROJECT] across the standard
pillars (Operational Excellence, Security, Reliability, Performance Efficiency, Cost
Optimization, Sustainability).
OBJECTIVE: Identify specific, evidenced gaps per pillar and prioritize remediation by risk and
effort.
INPUTS: Workload architecture and current configuration, monitoring/cost data where available.
CONSTRAINTS: Flag a gap only where a specific configuration or design choice deviates from the
pillar's best practice — not a generic "consider improving X."
DECISION RULES: Prioritize HIGH for gaps with a plausible near-term failure/cost/security
consequence; MEDIUM/LOW otherwise.
OUTPUT CONTRACT: Table [Pillar | Finding | Evidence | Priority | Recommendation] grouped by
pillar, plus a top-5 priority remediation list across all pillars.
VALIDATION: Each finding cites the specific configuration/design element examined, not a generic
pillar restatement.
ESCALATION: HIGH-priority Security or Reliability findings route to immediate remediation
planning, not the next quarterly cycle.
```

Output: Pillar-by-pillar Well-Architected findings with a prioritized top-5 remediation list.

Validate: Each finding cites the specific configuration/design element examined.

Next: HIGH findings feed immediate remediation; track via TDM-09-style debt tracking.

LEVEL L2 HUMAN GATE No

CA-06 Cloud Cost Optimization Analysis

Use when: Cloud spend needs to be reduced or brought under control without harming performance/reliability.

Outcome: A cost optimization plan quantifying savings per lever, with risk to performance/reliability stated.

Inputs: Current cloud billing/usage data, [THRESHOLD] savings target, [RISK_TOLERANCE]

PROMPT

```
CONTEXT: Analyzing cloud cost optimization opportunities for [SYSTEMS] on [PROJECT], targeting
[THRESHOLD] savings.
OBJECTIVE: Identify specific cost levers (rightsizing, reserved capacity, storage tiering, idle
resource elimination) with quantified savings and stated risk.
INPUTS: Current billing/usage data by service, utilization metrics.
CONSTRAINTS: Every recommended cost action must state its impact on performance or reliability,
even if "none" — do not present savings without a risk statement.
DECISION RULES: Recommend rightsizing where utilization is consistently below 40% of provisioned
capacity; recommend reserved/committed capacity where usage is stable and predictable over 12+
months; flag idle resources with zero activity in the analysis period for elimination.
OUTPUT CONTRACT: Table [Lever | Affected Resources | Estimated Monthly Savings | Risk to
Performance/Reliability | Implementation Effort] + total estimated savings vs [THRESHOLD].
VALIDATION: Savings figures recalculated from the actual billing/usage data provided, not
estimated generically.
ESCALATION: Actions affecting production-critical workloads require change-control sign-off
before implementation.
```

Output: Quantified cost optimization plan with risk-to-reliability stated per lever.

Validate: Savings figures recalculated from actual provided billing/usage data.

Next: Production-affecting actions route through standard change control.

LEVEL L1 HUMAN GATE Conditional

CA-07 Resilience / HA Design

Use when: Designing high-availability architecture for a workload with a specific uptime target.

Outcome: An HA design meeting the stated availability target with the specific redundancy mechanism per tier.

Inputs: Workload architecture, target availability in [SUCCESS_CRITERIA], [CONSTRAINTS]

PROMPT

```
CONTEXT: Designing high-availability architecture for <workload> on [PROJECT], target
availability [SUCCESS_CRITERIA].
OBJECTIVE: Define the redundancy mechanism per architectural tier needed to meet the stated
availability target.
INPUTS: Current/proposed architecture by tier, target availability, budget constraints.
CONSTRAINTS: Calculate composite availability across tiers (see SA-07 method) to confirm the
design actually meets the target — do not assume redundancy at one tier is sufficient without
checking the full chain.
DECISION RULES: Apply multi-AZ/multi-zone redundancy at minimum for any tier in the critical
path; apply multi-region only where the target availability or [RISK_TOLERANCE] cannot be met by
multi-AZ alone.
OUTPUT CONTRACT: Table [Tier | Redundancy Mechanism | Tier Availability | Composite Calculation]
+ confirmation the composite meets [SUCCESS_CRITERIA] + cost implication.
VALIDATION: Composite availability calculation shown and checked against the stated target.
ESCALATION: Designs unable to meet target within [CONSTRAINTS] escalate to sponsor for target or
budget renegotiation.
```

Output: HA design with calculated composite availability confirmed against target.

Validate: Composite availability calculation shown and checked against the target.

Next: Pair with CA-08 DR Strategy for full continuity coverage.

LEVEL L2 HUMAN GATE No

CA-08 DR Strategy Design

Use when: Defining disaster recovery approach and validating it meets RTO/RPO targets.

Outcome: A DR strategy with a specific pattern (backup/restore, pilot light, warm standby, active-active) justified against RTO/RPO.

Inputs: Target RTO/RPO in [SUCCESS_CRITERIA], workload criticality, [CONSTRAINTS] (budget)

PROMPT

```
CONTEXT: Designing disaster recovery strategy for <workload> on [PROJECT], target RTO/RPO in
[SUCCESS_CRITERIA].
OBJECTIVE: Select and justify a DR pattern that meets the stated RTO/RPO within budget
constraints.
INPUTS: Target RTO/RPO, workload criticality, current architecture, budget/[CONSTRAINTS].
CONSTRAINTS: Do not select a DR pattern by cost alone without checking it actually meets the
stated RTO/RPO — show the expected recovery time/data loss for the pattern chosen.
DECISION RULES: Backup/restore only if RTO tolerance is measured in hours-to-days; pilot
light/warm standby for RTO in tens of minutes to hours; active-active only if RTO/RPO approach
near-zero and budget supports it.
OUTPUT CONTRACT: Sections — Target RTO/RPO | Pattern Selected | Expected Actual RTO/RPO for this
Pattern | Cost Tier | Testing Cadence Recommended.
VALIDATION: Confirm the expected actual RTO/RPO for the chosen pattern is within the stated
target, not merely "in the right direction."
ESCALATION: Patterns that cannot meet target RTO/RPO within budget escalate to sponsor for target
or budget renegotiation.
```

Output: DR pattern selection with expected RTO/RPO checked against the target.

Validate: Expected RTO/RPO for the chosen pattern confirmed to meet the stated target.

Next: Schedule DR test per recommended cadence; feed into TL-09 Reliability Assessment.

LEVEL L2 HUMAN GATE Yes

CA-09 IAM & Security Posture Review

Use when: Reviewing cloud identity and access management configuration for excessive privilege or gaps.

Outcome: An IAM review flagging excessive privilege and missing controls, with least-privilege remediation.

Inputs: Current IAM policies/roles, access patterns if available, [CONSTRAINTS]

PROMPT

```
CONTEXT: Reviewing IAM posture for [SYSTEMS] on [PROJECT].
OBJECTIVE: Identify excessive privilege, unused permissions and missing controls (MFA, key
rotation) against least-privilege principles.
INPUTS: Current IAM roles/policies, access/usage logs if available.
CONSTRAINTS: Flag EXCESSIVE PRIVILEGE only where actual usage data (if available) or policy
structure shows broader access than the role's function requires — do not flag every wildcard
permission without functional context.
DECISION RULES: Flag CRITICAL if a role has account-admin-equivalent privilege without documented
justification, or if root/owner-level credentials lack MFA.
OUTPUT CONTRACT: Table [Role/Identity | Current Privilege | Actual Usage (if known) | Flag |
Recommended Least-Privilege Change].
VALIDATION: CRITICAL flags checked against the stated admin-equivalent/MFA-gap criteria
specifically.
ESCALATION: CRITICAL findings route to Security Architect and require immediate remediation, not
next-cycle scheduling.
```

Output: Least-privilege IAM review with critical findings flagged for immediate action.

Validate: Critical flags checked against the specific admin-equivalent/MFA-gap criteria.

Next: Critical items route to SEC-03 IAM Design Review for structural fix.

LEVEL L1 HUMAN GATE Yes

CA-10 Observability Strategy Design

Use when: Designing or improving monitoring, logging and alerting coverage for cloud workloads.

Outcome: An observability strategy with coverage mapped to failure modes, avoiding alert fatigue.

Inputs: [SYSTEMS] architecture, known failure modes/past incidents, current monitoring coverage

PROMPT

```
CONTEXT: Designing observability strategy for [SYSTEMS] on [PROJECT].
OBJECTIVE: Map monitoring/logging/alerting coverage to actual failure modes and past incidents,
avoiding both blind spots and alert fatigue.
INPUTS: Architecture components, past incident/failure history, current monitoring/alerting
configuration.
CONSTRAINTS: Every proposed alert must map to a specific, actionable failure mode with a defined
response — reject alerts that exist "just in case" with no clear action.
DECISION RULES: Flag COVERAGE GAP if a past incident's root cause had no corresponding
alert/metric at the time. Flag ALERT FATIGUE RISK if a proposed alert's threshold would fire more
than a stated acceptable frequency without action being needed.
OUTPUT CONTRACT: Table [Component | Failure Mode | Metric/Log Signal | Alert Threshold | Response
Action | Coverage Status].
VALIDATION: Confirm every past-incident root cause is checked against current coverage for the
gap flag.
ESCALATION: Recurring past-incident coverage gaps route to OPS-06 Observability Strategy for
platform-level implementation.
```

Output: Failure-mode-mapped observability strategy avoiding alert fatigue.

Validate: Past-incident root causes checked against current coverage for gaps.

Next: Implement via OPS-06; validate coverage after next incident.

LEVEL L2 HUMAN GATE No

Infrastructure Architect (IA)

Capacity, network, compute, storage and hybrid infrastructure design grounded in measured demand.

IA-01 Capacity Planning Analysis

Use when: You need a forward-looking capacity plan based on actual growth trends, not a flat estimate.

Outcome: A capacity plan showing when current infrastructure will breach threshold, with lead time to act.

Inputs: Current utilization trend data, growth projection, [THRESHOLD] for headroom

PROMPT

```
CONTEXT: Planning infrastructure capacity for [SYSTEMS] on [PROJECT] against [THRESHOLD] headroom
policy.
OBJECTIVE: Project when current capacity will breach threshold based on actual trend data, and
the lead time needed to act.
INPUTS: Historical utilization trend (compute/storage/network), growth projection basis,
procurement/provisioning lead time.
CONSTRAINTS: Project from actual historical trend data (linear or stated growth model) — do not
assume flat or unstated growth.
DECISION RULES: Flag ACT NOW if projected breach date is within the procurement/provisioning lead
time; otherwise state the safe monitoring window.
OUTPUT CONTRACT: Table [Resource | Current Utilization | Growth Rate | Projected Breach Date |
Lead Time Needed | Flag] + recommendation.
VALIDATION: Projected breach date recalculated from the stated growth rate and current
utilization, not asserted.
ESCALATION: ACT NOW items route to procurement/budget owner immediately given lead-time exposure.
```

Output: Trend-based capacity breach projection with lead-time-aware action flag.

Validate: Breach date recalculated from stated growth rate and current utilization.

Next: ACT NOW items go straight to procurement/budget request.

LEVEL L1 HUMAN GATE No

IA-02 Network Architecture Review

Use when: Reviewing network design for bottlenecks, single points of failure or segmentation gaps.

Outcome: A network review identifying specific bottlenecks and segmentation/security gaps.

Inputs: Network topology diagram, traffic patterns/volume, [SYSTEMS] segmentation requirements

PROMPT

```
CONTEXT: Reviewing network architecture for [SYSTEMS] on [PROJECT].
OBJECTIVE: Identify bandwidth bottlenecks, redundancy gaps and segmentation issues against actual
traffic patterns and security requirements.
INPUTS: Network topology, traffic volume/pattern data, required segmentation (e.g., PCI zone,
prod/non-prod isolation).
CONSTRAINTS: Flag a link as BOTTLENECK only where actual/projected traffic approaches its
provisioned capacity — do not flag based on topology complexity alone.
DECISION RULES: Flag SEGMENTATION GAP if traffic between zones requiring isolation is not
filtered/inspected by policy at the boundary.
OUTPUT CONTRACT: Table [Link/Zone | Issue Type (Bottleneck/Redundancy/Segmentation) | Evidence |
Severity | Recommendation].
VALIDATION: Bottleneck flags checked against actual traffic-to-capacity ratio, not visual
complexity.
ESCALATION: Segmentation gaps affecting compliance-scoped zones route to Security Architect
immediately.
```

Output: Evidence-based network issue list across bottlenecks, redundancy and segmentation.

Validate: Bottleneck flags checked against actual traffic-to-capacity data.

Next: Segmentation gaps route to SEC-02 Security Architecture Review.

LEVEL L2 HUMAN GATE No

IA-03 Compute Sizing Analysis

Use when: Sizing compute for a workload based on actual/expected demand rather than round-number defaults.

Outcome: A right-sized compute recommendation with the sizing calculation shown.

Inputs: Workload profile (CPU/memory/IO characteristics), expected concurrency/load, [THRESHOLD] headroom

PROMPT

```
CONTEXT: Sizing compute for <workload> on [PROJECT], targeting [THRESHOLD] headroom over expected
peak load.
OBJECTIVE: Recommend compute size (vCPU/memory/instance family) with the sizing calculation
shown.
INPUTS: Workload resource profile (from testing or comparable systems), expected peak
concurrency/load.
CONSTRAINTS: Base the recommendation on a stated per-unit resource consumption figure (e.g., "X
MB per concurrent session") — do not recommend a size without showing the derivation.
DECISION RULES: Required capacity = (expected peak load × per-unit consumption) × (1 +
[THRESHOLD] headroom); round up to the nearest standard instance/size tier.
OUTPUT CONTRACT: Sizing calculation shown step by step + recommended instance/size tier + cost
estimate + note on when to re-evaluate (e.g., load growth trigger).
VALIDATION: Recompute the final tier recommendation from the shown per-unit figures to confirm
arithmetic.
ESCALATION: If per-unit consumption data is unavailable, flag DATA GAP and recommend a load test
before sizing commitment.
```

Output: Calculated, right-sized compute recommendation with a re-evaluation trigger.

Validate: Final size tier recomputed from the shown per-unit consumption figures.

Next: Feed into IA-01 Capacity Planning for ongoing headroom tracking.

LEVEL L1 HUMAN GATE No

IA-04 Storage Architecture Review

Use when: Reviewing storage tiering, performance and cost fit against actual access patterns.

Outcome: A storage review matching tier/type to actual access pattern, flagging cost or performance mismatch.

Inputs: Storage inventory, access pattern data (frequency, size, IOPS needs), [CONSTRAINTS]

PROMPT

```
CONTEXT: Reviewing storage architecture for [SYSTEMS] on [PROJECT].
OBJECTIVE: Match storage tier/type to actual access pattern, flagging mismatches that waste cost
or harm performance.
INPUTS: Storage inventory by type/tier, access frequency and IOPS/throughput data per dataset.
CONSTRAINTS: Classify access pattern from actual data (hot/warm/cold based on access frequency),
not from data age assumption alone.
DECISION RULES: Flag COST MISMATCH if cold/infrequently-accessed data sits on a high-
performance/high-cost tier. Flag PERFORMANCE MISMATCH if high-IOPS data sits on a tier that
cannot meet its throughput needs.
OUTPUT CONTRACT: Table [Dataset | Current Tier | Access Pattern (Hot/Warm/Cold, with basis) |
Recommended Tier | Mismatch Type | Estimated Impact].
VALIDATION: Access pattern classification checked against the actual frequency/IOPS data
provided.
ESCALATION: Performance mismatches on production-critical datasets route to immediate remediation
planning.
```

Output: Storage tier/access-pattern mismatch review with cost and performance impact.

Validate: Hot/warm/cold classification checked against actual access data.

Next: Cost mismatches feed CA-06 Cloud Cost Optimization if cloud-hosted.

LEVEL L1 HUMAN GATE No

IA-05 HA/DR Design Review

Use when: Reviewing an existing infrastructure HA/DR design for gaps against stated targets.

Outcome: A gap review of HA/DR design against stated RTO/RPO, with specific missing mechanisms named.

Inputs: Current HA/DR architecture and procedures, target RTO/RPO in [SUCCESS_CRITERIA]

PROMPT

```
CONTEXT: Reviewing HA/DR design for [SYSTEMS] on [PROJECT] against target RTO/RPO in
[SUCCESS_CRITERIA].
OBJECTIVE: Identify specific gaps between the current design/procedures and the stated recovery
targets.
INPUTS: Current HA/DR architecture, backup frequency/retention, last DR test results if any.
CONSTRAINTS: Assess RPO gap using actual backup/replication frequency, not the target — e.g.,
nightly backups cannot meet a 1-hour RPO regardless of design intent elsewhere.
DECISION RULES: Flag RPO GAP if backup/replication frequency exceeds the target RPO window. Flag
UNTESTED if no DR test has been run within the last 12 months.
OUTPUT CONTRACT: Table [Component | Current Mechanism | Target | Actual Capability | Gap |
Recommendation].
VALIDATION: RPO gap calculated directly from stated backup/replication frequency vs target.
ESCALATION: Untested DR procedures for critical systems route to a scheduled DR exercise within
the current quarter.
```

Output: RTO/RPO gap review with specific missing mechanisms and a testing status flag.

Validate: RPO gap calculated directly from actual backup/replication frequency vs target.

Next: Schedule DR test; feed gaps into CA-08 DR Strategy Design.

LEVEL L2 HUMAN GATE No

IA-06 Backup Strategy Review

Use when: Reviewing whether current backup policy actually protects against realistic failure/loss scenarios.

Outcome: A backup strategy review testing coverage against specific loss scenarios, not just 'backups exist.'

Inputs: Current backup policy (frequency, retention, location), [SYSTEMS] criticality, target RPO

PROMPT

```
CONTEXT: Reviewing backup strategy for [SYSTEMS] on [PROJECT].
OBJECTIVE: Test whether current backup policy protects against specific realistic loss scenarios
(accidental deletion, ransomware, regional outage, corruption).
INPUTS: Backup frequency/retention/location policy, criticality of the data, target RPO.
CONSTRAINTS: Evaluate each scenario independently — a policy that protects against accidental
deletion may not protect against ransomware (e.g., if backups are not immutable/air-gapped).
DECISION RULES: Flag VULNERABLE to ransomware if backups are writable/deletable from the same
credentials as production. Flag VULNERABLE to regional outage if backups are stored only in the
same region as production.
OUTPUT CONTRACT: Table [Scenario | Current Protection | Vulnerable? (Y/N) | Recommendation] +
overall backup posture rating.
VALIDATION: Confirm each scenario's vulnerability flag is checked against the specific stated
policy detail (location, credential isolation), not general backup existence.
ESCALATION: Ransomware-vulnerable policies on critical systems route to immediate remediation,
given active threat landscape.
```

Output: Scenario-tested backup strategy review with specific vulnerability flags.

Validate: Each vulnerability flag checked against the specific policy detail, not general existence.

Next: Ransomware-vulnerable findings escalate for immediate remediation.

LEVEL L1 HUMAN GATE Yes

IA-07 Migration Readiness Assessment

Use when: Before a physical or platform migration, assessing whether infrastructure/data/dependencies are actually ready.

Outcome: A readiness assessment naming specific blockers, not a general 'looks mostly ready' impression.

Inputs: Migration scope, dependency inventory, current infrastructure state, target environment specs

PROMPT

```
CONTEXT: Assessing migration readiness for <infrastructure/workload> to <target environment> on
[PROJECT].
OBJECTIVE: Identify specific technical, dependency and data blockers preventing migration, and
confirm what is genuinely ready.
INPUTS: Migration scope, dependency inventory (upstream/downstream systems), current vs target
environment specifications.
CONSTRAINTS: A dependency is READY only if it has been explicitly validated compatible with the
target environment — do not assume compatibility from version number alone.
DECISION RULES: Flag BLOCKING if a dependency has no validated compatibility path to the target;
flag NEEDS VALIDATION if compatibility is plausible but untested.
OUTPUT CONTRACT: Table [Item | Current State | Target Requirement | Status (Ready/Needs
Validation/Blocking) | Action Needed].
VALIDATION: Confirm every READY status has an explicit validation record, not an assumption.
ESCALATION: BLOCKING items route to the migration planning team before a cutover date is
committed.
```

Output: Specific, evidence-based migration readiness assessment with named blockers.

Validate: Ready statuses confirmed to have an explicit validation record.

Next: Blocking items resolved before CA-02 wave scheduling commits a date.

LEVEL L1 HUMAN GATE No

IA-08 Hybrid Infrastructure Design

Use when: Designing infrastructure spanning on-premises and cloud, with explicit workload placement rationale.

Outcome: A hybrid design with each workload's placement (on-prem vs cloud) explicitly justified.

Inputs: Workload inventory, [CONSTRAINTS] (data residency, latency, existing investment), connectivity requirements

PROMPT

```
CONTEXT: Designing hybrid infrastructure for [PROGRAM] spanning on-premises and cloud.
OBJECTIVE: Justify each workload's placement (on-prem, cloud, or split) against data residency,
latency and existing investment constraints.
INPUTS: Workload inventory, data residency/regulatory constraints, latency-sensitive
dependencies, existing on-prem investment/amortization status.
CONSTRAINTS: Placement decisions must cite the specific constraint driving them (residency law,
latency SLA, sunk investment) — not general "keep some on-prem for now."
DECISION RULES: Keep ON-PREM if a hard regulatory/residency constraint applies or latency SLA
cannot be met from cloud; otherwise favor CLOUD unless sunk-investment payback period justifies
delay.
OUTPUT CONTRACT: Table [Workload | Placement | Constraint Driving Decision | Connectivity
Requirement (VPN/Direct Connect/etc.)] + network connectivity design summary.
VALIDATION: Every placement decision cites a specific named constraint, not a general preference.
ESCALATION: Residency/regulatory-driven placements route to legal/compliance for confirmation.
```

Output: Constraint-justified hybrid workload placement design with connectivity plan.

Validate: Every placement decision cites a specific named constraint.

Next: Regulatory placements confirmed with legal/compliance before build.

LEVEL L2 HUMAN GATE Yes

Enterprise Architect (EA)

Capability mapping, portfolio rationalization and transformation roadmaps at the enterprise level.

EA-01 Capability Mapping

Use when: Building or updating a business capability map to align technology investment with business function.

Outcome: A capability map with maturity/health rating per capability, grounded in evidence, not aspiration.

Inputs: Business function inventory, [SOURCE_DATA] on current supporting systems/processes

PROMPT

```
CONTEXT: Mapping business capabilities for <business unit/domain> under [PROGRAM].
OBJECTIVE: Produce a capability map with a maturity/health rating per capability based on actual supporting system and process evidence.
INPUTS: Business function inventory, supporting systems/processes per function, known pain points.
CONSTRAINTS: Rate maturity from evidence (system age/fit, process manual-ness, known incidents) — not from stakeholder opinion of importance.
DECISION RULES: Rate LOW maturity if the capability relies on manual workarounds or end-of-life systems; HIGH if supported by modern, fit-for-purpose systems with low incident history.
OUTPUT CONTRACT: Table [Capability | Supporting Systems | Maturity Rating | Evidence | Strategic Importance | Investment Priority Flag].
VALIDATION: Maturity ratings checked against the cited system/process evidence, not general reputation.
ESCALATION: Capabilities rated LOW maturity + HIGH strategic importance route to EA-06 Transformation Roadmap as priority candidates.
```

Output: Evidence-based capability map with maturity ratings and investment priority flags.

Validate: Maturity ratings checked against cited system/process evidence.

Next: High-priority gaps feed EA-06 Transformation Roadmap.

LEVEL L2 HUMAN GATE No

EA-02 Application Portfolio Rationalization

Use when: The application landscape has redundancy or bloat and needs a rationalization (TIME model) pass.

Outcome: A TIME-classified application portfolio (Tolerate/Invest/Migrate/Eliminate) with clear rationale.

Inputs: Application inventory, usage/cost data, capability map (EA-01), [CONSTRAINTS]

PROMPT

```
CONTEXT: Rationalizing the application portfolio for [CUSTOMER]/[PROGRAM] using the TIME model.
OBJECTIVE: Classify each application as Tolerate, Invest, Migrate or Eliminate, with business and technical rationale.
INPUTS: Application inventory, cost/usage data, business capability fit, technical health (age, support status, vendor viability).
CONSTRAINTS: Classification must weigh both business fit (does it serve a needed capability well) and technical health — do not classify on cost alone.
DECISION RULES: ELIMINATE if business fit is low AND a better alternative exists; INVEST if business fit is high AND technical health is low; MIGRATE if fit is high but hosted on an end-of-life platform; TOLERATE if fit and health are both adequate for now.
OUTPUT CONTRACT: Table [Application | Business Fit | Technical Health | TIME Classification | Rationale | Estimated Cost Impact of Action].
VALIDATION: Each classification checked against both the business-fit and technical-health inputs, not one alone.
ESCALATION: ELIMINATE candidates with active users route to business owner confirmation before decommission planning.
```

Output: TIME-classified application portfolio with dual business/technical rationale.

Validate: Classifications checked against both business-fit and technical-health inputs.

Next: Eliminate/migrate candidates feed EA-06 Transformation Roadmap.

LEVEL L2 HUMAN GATE Yes

EA-03 Technology Standards Assessment

Use when: Assessing compliance with, or need to update, enterprise technology standards.

Outcome: A standards compliance assessment with specific deviations and their business/risk justification (if any).

Inputs: Current technology standards catalog, actual technology usage across [SYSTEMS]

PROMPT

```
CONTEXT: Assessing technology standards compliance across [SYSTEMS] for [PROGRAM].
OBJECTIVE: Identify specific deviations from enterprise standards and whether each has a
documented business justification.
INPUTS: Standards catalog (approved technologies/versions), actual technology inventory in use.
CONSTRAINTS: A deviation is JUSTIFIED only if a documented exception/waiver exists — do not
assume justification from "it's been working fine."
DECISION RULES: Flag UNJUSTIFIED DEVIATION if no waiver exists; flag STANDARD GAP if a widely-
used, sound technology has no corresponding standard entry (may indicate the standard needs
updating, not the usage).
OUTPUT CONTRACT: Table [Technology in Use | Standard Status (Compliant/Deviation/Gap) | Waiver on
File? | Recommendation (Remediate/Formalize Waiver/Update Standard)].
VALIDATION: Confirm deviation flags checked against the actual waiver record, not assumed intent.
ESCALATION: Unjustified deviations in production-critical systems route to architecture review
board for a remediation or waiver decision.
```

Output: Standards compliance assessment distinguishing justified deviations from gaps.

Validate: Deviation flags checked against actual waiver records on file.

Next: Unjustified deviations route to architecture review board.

LEVEL L1 HUMAN GATE Conditional

EA-04 Target Architecture Definition

Use when: Defining the future-state enterprise architecture to guide multi-year investment.

Outcome: A target architecture definition connected explicitly to business strategy, not a generic reference model.

Inputs: Business strategy/[BUSINESS_OUTCOME], current-state architecture, capability map (EA-01)

PROMPT

```
CONTEXT: Defining target architecture for <domain> in service of [BUSINESS_OUTCOME] under
[PROGRAM].
OBJECTIVE: Define the future-state architecture with each major element explicitly tied to a
business strategy driver.
INPUTS: Business strategy/outcome statements, current-state architecture, capability map.
CONSTRAINTS: Every target-state element must trace to a specific business driver or capability
gap — reject generic "modernize everything" statements without a traced rationale.
DECISION RULES: Prioritize target-state elements that close HIGH-priority capability gaps (from
EA-01) first in the narrative, even if full target state covers more.
OUTPUT CONTRACT: Sections — Business Drivers | Target Architecture by Domain (data, application,
technology, integration) | Traceability to Capability Gaps | Key Principles Applied.
VALIDATION: Spot-check 3 target-state elements to confirm each traces to a named business driver
or capability gap.
ESCALATION: Target architecture with major cost/organizational implications routes to executive
sponsor for endorsement before roadmap work begins.
```

Output: Business-traced target architecture definition across key domains.

Validate: Sampled target-state elements confirmed traced to named business drivers.

Next: Feed into EA-06 Transformation Roadmap for sequencing.

LEVEL L2 HUMAN GATE Yes

EA-05 Architecture Principles Development

Use when: Establishing or refreshing enterprise architecture principles to guide decentralized decisions.

Outcome: A concise set of testable architecture principles, each with a stated implication, not motherhood statements.

Inputs: [BUSINESS_OUTCOME], known past architecture decision conflicts/pain points

PROMPT

```
CONTEXT: Developing architecture principles for [PROGRAM]/[CUSTOMER] to guide decentralized
architecture decisions.
OBJECTIVE: Define a concise principle set, each testable and with a stated practical implication
— not vague statements everyone would agree with.
INPUTS: Business outcome priorities, known past conflicts where a principle would have helped
decide.
CONSTRAINTS: Reject any principle that cannot be used to say "no" to at least one plausible real
design choice — untestable principles add no value.
DECISION RULES: For each principle, require a stated "Implication" (what it means practitioners
must do differently) and a stated "Rationale" (why, tied to business outcome).
OUTPUT CONTRACT: Table [Principle | Rationale | Implication | Example Decision It Would Resolve].
VALIDATION: Confirm every principle's "Example Decision" is a concrete, plausible scenario, not
abstract.
ESCALATION: Principles with organization-wide behavioral implications route to executive sponsor
for endorsement before publication.
```

Output: Testable architecture principles with stated implications and worked examples.

Validate: Each principle's example decision confirmed concrete and plausible.

Next: Publish and apply in SA-01/CA-01 architecture option evaluations.

LEVEL L1 HUMAN GATE Yes

EA-06 Transformation Roadmap

Use when: Sequencing multi-year architecture/capability change into an executable roadmap.

Outcome: A phased transformation roadmap sequenced by dependency and value, with named milestones.

Inputs: Target architecture (EA-04), capability gaps (EA-01), [CONSTRAINTS] (budget, org capacity)

PROMPT

```
CONTEXT: Building the transformation roadmap toward the target architecture for [PROGRAM].
OBJECTIVE: Sequence the transformation into phases based on dependency, value realization speed
and organizational capacity.
INPUTS: Target architecture, prioritized capability gaps, budget/capacity constraints.
CONSTRAINTS: Do not sequence purely by "logical" architecture layering if it delays all business
value to the final phase — surface value earlier where dependency allows.
DECISION RULES: Phase 1 = capability gaps that are both HIGH priority (EA-01) and dependency-
independent; later phases follow the dependency chain.
OUTPUT CONTRACT: Phased table [Phase | Capability/Architecture Element | Dependency | Value
Delivered | Estimated Duration | Investment Tier].
VALIDATION: Confirm Phase 1 items are genuinely dependency-independent, not assumed so.
ESCALATION: Roadmap requiring investment beyond current budget cycle routes to executive sponsor
for multi-year funding approval.
```

Output: Phased, dependency-checked transformation roadmap with named value per phase.

Validate: Phase 1 items confirmed genuinely dependency-independent.

Next: Fund via multi-year business case; track via PMO-05.

LEVEL L2 HUMAN GATE Yes

EA-07 Modernization Business Case

Use when: A modernization initiative needs a business case to secure funding.

Outcome: A business case quantifying cost of inaction vs investment required, not just a technical justification.

Inputs: Modernization scope, current-state cost/risk data, [BUSINESS_OUTCOME]

PROMPT

```
CONTEXT: Building the business case for modernizing <capability/system> under [PROGRAM],
targeting [BUSINESS_OUTCOME].
OBJECTIVE: Quantify the cost of inaction (ongoing risk, maintenance cost, opportunity cost)
against the investment required to modernize.
INPUTS: Current-state operating cost, incident/risk history, modernization investment estimate,
expected benefits.
CONSTRAINTS: Cost of inaction must be quantified from actual current-state data (support costs,
incident costs, lost opportunity) — not asserted qualitatively.
DECISION RULES: Present a payback period or breakeven point explicitly; do not present investment
cost without a corresponding return timeline.
OUTPUT CONTRACT: Sections — Current State Cost/Risk (quantified) | Modernization Investment
Required | Expected Benefits (quantified where possible) | Payback Period | Risk of Inaction if
Deferred.
VALIDATION: Confirm the payback period is calculated from the stated investment and benefit
figures, not asserted.
ESCALATION: Business cases requiring capital investment approval route to finance/executive
sponsor per governance thresholds.
```

Output: Quantified modernization business case with a calculated payback period.

Validate: Payback period recalculated from stated investment and benefit figures.

Next: Route through PMO-08 Executive Portfolio Report for funding approval.

LEVEL L2 HUMAN GATE Yes

EA-08 Architecture Governance Review

Use when: Reviewing whether the architecture governance process itself is effective and being followed.

Outcome: A governance effectiveness review distinguishing process design gaps from adherence gaps.

Inputs: Governance process documentation, evidence of recent decisions/exceptions, [SOURCE_DATA]

PROMPT

```
CONTEXT: Reviewing architecture governance effectiveness for [PROGRAM]/[CUSTOMER].
OBJECTIVE: Distinguish whether governance issues stem from process design gaps or from non-
adherence to an adequate process.
INPUTS: Governance process/charter, recent decision records, instances of bypassed or exception-
driven decisions.
CONSTRAINTS: Classify each observed issue precisely — DESIGN GAP (the process doesn't cover this
case) vs ADHERENCE GAP (the process covers it but wasn't followed).
DECISION RULES: DESIGN GAP if no defined process step exists for the observed scenario; ADHERENCE
GAP if a defined step was bypassed with no documented exception.
OUTPUT CONTRACT: Table [Observed Issue | Classification | Evidence | Recommended Fix (Process
Change vs Compliance Action)].
VALIDATION: Confirm each classification is checked against the actual governance charter's
defined coverage.
ESCALATION: Adherence gaps involving repeated bypass by the same team route to executive sponsor
for a compliance conversation.
```

Output: Governance effectiveness review distinguishing design gaps from adherence gaps.

Validate: Classifications checked against the actual governance charter coverage.

Next: Design gaps update the governance charter; adherence gaps route to sponsor.

LEVEL L2 HUMAN GATE Yes

Security Architect (SEC)

Threat modeling, controls and compliance grounded in evidence; consequential outputs always human-gated.

SEC-01 Threat Modeling

Use when: A new system or significant change needs structured threat identification before build.

Outcome: A structured threat model (e.g., STRIDE-based) with specific, evidenced threats and mitigations.

Inputs: System architecture/data flow diagram, data classification, trust boundaries

PROMPT

```
CONTEXT: Threat modeling <system/feature> on [PROJECT] handling data classified as
<classification>.
OBJECTIVE: Identify specific threats per trust boundary using a structured method (e.g., STRIDE)
with proposed mitigations.
INPUTS: Architecture/data flow diagram with trust boundaries marked, data classification, known
entry points.
CONSTRAINTS: Identify threats specific to this system's actual data flows and boundaries — do not
produce a generic threat checklist disconnected from the diagram.
DECISION RULES: Rate threat severity by (likelihood × impact on the classified data); prioritize
mitigations for threats crossing external trust boundaries first.
OUTPUT CONTRACT: Table [Trust Boundary | Threat (STRIDE category) | Attack Vector | Likelihood |
Impact | Severity | Mitigation | Residual Risk].
VALIDATION: Confirm every threat entry references a specific boundary/flow from the diagram, not
a generic entry.
ESCALATION: HIGH severity threats with no feasible mitigation route to sponsor for risk
acceptance or scope change — human sign-off required.
```

Output: Diagram-grounded threat model with severity-rated mitigations and residual risk.

Validate: Each threat entry confirmed to reference a specific diagram boundary/flow.

Next: Feed into SEC-02 Security Architecture Review for controls design.

LEVEL L2 HUMAN GATE Yes

SEC-02 Security Architecture Review

Use when: Reviewing a proposed or existing security architecture for control completeness.

Outcome: A security architecture review mapping controls to identified threats, flagging uncovered ones.

Inputs: Threat model (SEC-01), proposed/current security controls, [CONSTRAINTS]

PROMPT

```
CONTEXT: Reviewing security architecture for <system> on [PROJECT] against the threat model.
OBJECTIVE: Confirm every HIGH/MEDIUM severity threat has a corresponding control; flag any
without one.
INPUTS: Threat model output, current/proposed security controls (network, application, data,
identity layers).
CONSTRAINTS: A control counts as coverage only if it directly addresses the specific attack
vector in the threat model — a general "firewall exists" does not cover an application-layer
threat.
DECISION RULES: Flag UNCOVERED if a HIGH/MEDIUM severity threat has no directly-mapped control.
Flag OVER-CONTROLLED only where a control adds material cost/friction with no corresponding
threat (for cost-conscious review).
OUTPUT CONTRACT: Table [Threat (from SEC-01) | Mapped Control | Coverage Status | Gap Remediation
(if uncovered)].
VALIDATION: Confirm every "Covered" status shows the specific control-to-vector mapping, not a
general control category.
ESCALATION: Uncovered HIGH severity threats block production sign-off pending remediation or
documented risk acceptance.
```

Output: Threat-to-control coverage map with explicit gap remediation.

Validate: Covered statuses show the specific control-to-vector mapping.

Next: Uncovered HIGH threats block production sign-off until resolved.

LEVEL L2 HUMAN GATE Yes

SEC-03 IAM Design Review

Use when: Reviewing identity and access management design for structural least-privilege compliance.

Outcome: An IAM design review confirming role-based access maps correctly to least-privilege need.

Inputs: IAM role/permission design, job function/access need mapping, [CONSTRAINTS]

PROMPT

```
CONTEXT: Reviewing IAM design for [SYSTEMS] on [PROJECT].
OBJECTIVE: Confirm each role's permissions map to documented job function need, flagging excess.
INPUTS: Role definitions with permissions, documented job function access requirements.
CONSTRAINTS: Flag excess only where a permission has no traceable link to a stated job function
requirement — do not flag based on permission count alone.
DECISION RULES: Flag CRITICAL if a non-privileged role includes admin-equivalent permissions;
flag REVIEW NEEDED if a permission's link to job function is unclear/undocumented.
OUTPUT CONTRACT: Table [Role | Permission | Linked Job Function (or none) | Flag |
Recommendation].
VALIDATION: Critical flags checked against the specific admin-equivalent permission list, not
general suspicion.
ESCALATION: CRITICAL findings route to immediate remediation before the design is approved for
build.
```

Output: Least-privilege IAM design review with critical/review-needed flags.

Validate: Critical flags checked against a specific admin-equivalent permission list.

Next: Critical items block design sign-off until remediated.

LEVEL L1 HUMAN GATE Yes

SEC-04 Zero Trust Readiness Assessment

Use when: Assessing how far current architecture is from Zero Trust principles, with a prioritized path.

Outcome: A Zero Trust readiness assessment against specific pillars, with a prioritized closing-the-gap plan.

Inputs: Current architecture/identity/network controls, Zero Trust pillar reference (identity, device, network, application, data)

PROMPT

```
CONTEXT: Assessing Zero Trust readiness for [SYSTEMS] on [PROJECT] against standard pillars
(Identity, Device, Network, Application, Data).
OBJECTIVE: Rate current maturity per pillar with evidence, and prioritize gap-closing actions by
risk reduction per effort.
INPUTS: Current identity verification approach, device trust posture, network segmentation,
application-layer controls, data protection controls.
CONSTRAINTS: Rate maturity from actual control evidence per pillar — do not rate a pillar HIGH
based on partial or aspirational controls.
DECISION RULES: Prioritize gap-closing actions by (risk reduction ÷ implementation effort);
pillars with implicit trust still assumed (e.g., "trusted network" perimeter model) get flagged
first.
OUTPUT CONTRACT: Table [Pillar | Current Maturity | Evidence | Gap | Priority] + prioritized
roadmap of top actions.
VALIDATION: Maturity ratings checked against the specific control evidence cited per pillar.
ESCALATION: Pillars rated lowest maturity with highest data sensitivity route to CISO/security
leadership for investment prioritization.
```

Output: Pillar-rated Zero Trust readiness assessment with a prioritized gap-closing roadmap.

Validate: Maturity ratings checked against the specific cited control evidence.

Next: Feed top actions into security investment planning.

LEVEL L2 HUMAN GATE No

SEC-05 Security Controls Gap Analysis

Use when: Assessing compliance/coverage against a specific control framework (e.g., NIST, ISO 27001, CIS).

Outcome: A control-by-control gap analysis against a named framework, distinguishing gap severity.

Inputs: Named control framework, current control implementation evidence, [CONSTRAINTS]

PROMPT

```
CONTEXT: Assessing security controls gap for [SYSTEMS] on [PROJECT] against <named framework>.
OBJECTIVE: Identify which framework controls are implemented, partially implemented, or missing,
with evidence.
INPUTS: Framework control list, current implementation evidence (configurations, policies,
procedures).
CONSTRAINTS: Rate a control IMPLEMENTED only with direct evidence (configuration, policy
document, log) — do not rate implemented based on stated intent alone.
DECISION RULES: Flag CRITICAL GAP if a missing/partial control corresponds to a framework-flagged
high-priority or foundational control; otherwise STANDARD GAP.
OUTPUT CONTRACT: Table [Control ID | Control Description | Status (Implemented/Partial/Missing) |
Evidence | Gap Severity | Remediation].
VALIDATION: Every IMPLEMENTED status checked against cited direct evidence, not stated intent.
ESCALATION: CRITICAL GAPs route to compliance/security leadership with a remediation timeline
requirement.
```

Output: Evidence-based control gap analysis against a named framework, severity-rated.

Validate: Implemented statuses checked against cited direct evidence.

Next: Critical gaps feed a remediation plan with tracked deadlines.

LEVEL L2 HUMAN GATE Conditional

SEC-06 Security Risk Assessment

Use when: A specific security risk needs formal assessment and treatment recommendation.

Outcome: A formal security risk assessment with likelihood/impact scoring and a treatment recommendation.

Inputs: The identified risk/vulnerability, affected [SYSTEMS], [RISK_TOLERANCE]

PROMPT

```
CONTEXT: Assessing security risk: <describe risk/vulnerability> affecting [SYSTEMS] on [PROJECT].
OBJECTIVE: Score the risk (likelihood × impact) and recommend a treatment
(mitigate/accept/transfer/avoid) against [RISK_TOLERANCE].
INPUTS: Risk/vulnerability description, affected assets/data classification, exploitability
evidence if known (e.g., CVSS score, active exploitation reports).
CONSTRAINTS: Use available exploitability evidence (CVSS, known exploit activity) to inform
likelihood — do not assign likelihood by intuition alone.
DECISION RULES: Recommend MITIGATE if risk score exceeds [RISK_TOLERANCE] and a feasible control
exists; ACCEPT only if within tolerance and documented by an accountable owner; TRANSFER if
insurable/contractable.
OUTPUT CONTRACT: Sections — Risk Description | Likelihood (with basis) | Impact (with basis) |
Score | Recommended Treatment | Residual Risk if Treated.
VALIDATION: Confirm likelihood rating cites specific exploitability evidence, not general
assumption.
ESCALATION: ACCEPT recommendations above a materiality threshold require named executive sign-off
— human decision required.
```

Output: Scored security risk assessment with a treatment recommendation and residual risk.

Validate: Likelihood rating cites specific exploitability evidence.

Next: Accept decisions require named executive sign-off before closing.

LEVEL L1 HUMAN GATE Yes

SEC-07 Vulnerability Prioritization

Use when: A vulnerability scan/report has too many findings to fix at once and needs risk-based triage.

Outcome: A prioritized vulnerability remediation list based on exploitability and exposure, not CVSS score alone.

Inputs: Vulnerability scan results, [SYSTEMS] exposure context (internet-facing vs internal), data sensitivity

PROMPT

```
CONTEXT: Prioritizing vulnerability remediation for [SYSTEMS] on [PROJECT] from recent scan
results.
OBJECTIVE: Prioritize findings by actual exploitability and exposure context, not CVSS score
alone.
INPUTS: Scan results with CVSS scores, exposure context (internet-facing, internal-only, air-
gapped), data sensitivity of affected systems, known active-exploitation status if available.
CONSTRAINTS: Do not prioritize purely by CVSS score — a critical CVSS on an air-gapped, low-
sensitivity system may rank below a medium CVSS on an internet-facing system handling sensitive
data.
DECISION RULES: Priority = function of (CVSS severity × exposure factor × data sensitivity ×
known active exploitation); internet-facing + actively exploited findings are always top priority
regardless of raw CVSS tier.
OUTPUT CONTRACT: Table [Finding | CVSS | Exposure | Data Sensitivity | Active Exploitation? |
Priority Rank | Remediation Deadline Recommendation].
VALIDATION: Confirm top-ranked findings include the exposure and exploitation context, not CVSS
score alone.
ESCALATION: Actively-exploited, internet-facing findings require same-day escalation to the
security operations/incident team.
```

Output: Context-weighted vulnerability priority list with remediation deadlines.

Validate: Top rankings confirmed to weigh exposure and exploitation, not CVSS alone.

Next: Actively-exploited findings escalate same-day to security operations.

LEVEL L1 HUMAN GATE Conditional

SEC-08 Compliance Mapping

Use when: Mapping a system/process to applicable regulatory or contractual compliance requirements.

Outcome: A compliance mapping showing which requirements apply, current status, and evidence gaps.

Inputs: Applicable regulation(s)/contract clauses, [SYSTEMS] scope, current control evidence

PROMPT

```
CONTEXT: Mapping compliance requirements for [SYSTEMS] on [PROJECT] against <named
regulation/contract>.
OBJECTIVE: Identify which specific requirements apply to this system's scope, current compliance
status, and evidence gaps.
INPUTS: Regulation/contract text or requirement list, system scope/data handled, current control
evidence.
CONSTRAINTS: A requirement is IN SCOPE only if the system's actual data handling or function
triggers it — do not assume all requirements apply blanket-wide.
DECISION RULES: Rate COMPLIANT only with direct evidence; rate GAP if no evidence or evidence is
stale/unverified; mark NOT APPLICABLE with explicit reasoning if scope doesn't trigger the
requirement.
OUTPUT CONTRACT: Table [Requirement | In Scope? (with reason) | Status (Compliant/Gap/N-A) |
Evidence | Remediation if Gap].
VALIDATION: Every COMPLIANT status checked against cited direct evidence, and every N-A checked
against a stated scope reason.
ESCALATION: GAP findings on legally mandated requirements route to legal/compliance for a
remediation deadline.
```

Output: Scope-justified compliance mapping with evidence-graded status per requirement.

Validate: Compliant and N-A statuses each checked against cited evidence or scope reasoning.

Next: Legal mandate gaps route to legal/compliance for a remediation deadline.

LEVEL L2 HUMAN GATE Yes

Data Architect (DA)

Data architecture, governance, quality and AI/RAG readiness grounded in evidence and lineage.

DA-01 Data Architecture Design

Use when: Designing the data architecture for a new domain or capability.

Outcome: A data architecture design with storage/processing pattern justified against data characteristics.

Inputs: Data characteristics (volume, velocity, variety), consumption patterns, [CONSTRAINTS]

PROMPT

```
CONTEXT: Designing data architecture for <domain/capability> on [PROJECT].
OBJECTIVE: Select and justify data storage/processing patterns (transactional, analytical,
streaming, hybrid) against actual data characteristics and consumption needs.
INPUTS: Data volume/velocity/variety profile, consumption patterns (real-time vs batch, query
complexity), constraints.
CONSTRAINTS: Justify the pattern against stated consumption needs — do not default to a data
warehouse or data lake pattern without checking fit.
DECISION RULES: Favor transactional stores for high-consistency operational needs; analytical
stores for aggregate/historical query patterns; streaming architecture only where genuine real-
time consumption exists.
OUTPUT CONTRACT: Sections — Data Profile Summary | Architecture Pattern Selected with Rationale |
Storage/Processing Component Map | Rejected Alternatives.
VALIDATION: Confirm the selected pattern's characteristics match the stated consumption
requirement, not convention.
ESCALATION: Cross-domain data sharing implications route to DA-03 Data Governance Framework for
policy alignment.
```

Output: Justified data architecture pattern selection with component map.

Validate: Selected pattern's characteristics checked against actual consumption needs.

Next: Cross-domain sharing routes to DA-03 Governance Framework.

LEVEL L2 HUMAN GATE No

DA-02 Data Modeling Review

Use when: Reviewing a proposed or existing data model for normalization, integrity and fit-for-purpose issues.

Outcome: A data model review flagging integrity risks and fit-for-purpose mismatches with the model's actual use case.

Inputs: Data model/schema, intended use case (transactional vs analytical), known query patterns

PROMPT

```
CONTEXT: Reviewing the data model for <domain/system> on [PROJECT], intended use:
<transactional/analytical>.
OBJECTIVE: Identify integrity risks (missing constraints, ambiguous relationships) and fit-for-
purpose mismatches against the model's actual intended use.
INPUTS: Data model/schema, intended use case, known/expected query patterns.
CONSTRAINTS: Evaluate normalization level against the stated use case — a highly normalized model
may be correct for transactional integrity but wrong for analytical query performance; do not
apply one standard universally.
DECISION RULES: Flag INTEGRITY RISK if a relationship lacks a defined constraint (foreign key,
cardinality) that the business rule requires. Flag PERFORMANCE MISMATCH if the normalization
level conflicts with the stated query pattern (e.g., highly normalized model for heavy analytical
aggregation).
OUTPUT CONTRACT: Table [Entity/Relationship | Issue Type | Evidence | Recommendation].
VALIDATION: Confirm normalization-level flags are checked against the stated use case, not a
universal "third normal form is always correct" assumption.
ESCALATION: Integrity risks affecting financial/regulatory data route to compliance review.
```

Output: Use-case-matched data model review with integrity and performance flags.

Validate: Normalization flags checked against the stated actual use case.

Next: Financial/regulatory integrity risks route to compliance review.

LEVEL L1 HUMAN GATE No

DA-03 Data Governance Framework

Use when: Establishing or maturing data governance — ownership, stewardship, quality accountability.

Outcome: A governance framework assigning explicit ownership and quality accountability per data domain.

Inputs: Data domain inventory, current ownership (if any), [CONSTRAINTS]

PROMPT

```
CONTEXT: Establishing data governance for <domain(s)> under [PROGRAM].
OBJECTIVE: Assign explicit data ownership, stewardship and quality accountability per domain,
avoiding ambiguous "everyone owns it" outcomes.
INPUTS: Data domain inventory, current ownership state (formal or informal), known
quality/consistency pain points.
CONSTRAINTS: Every domain must have exactly one accountable Data Owner — shared/ambiguous
ownership is treated as a gap to close, not an acceptable state.
DECISION RULES: Assign ownership to the business function that is the authoritative source of the
data (system of record owner), not the function that merely consumes it most.
OUTPUT CONTRACT: Table [Data Domain | System of Record | Data Owner (role) | Data Steward (role)
| Quality Accountability | Current Gap].
VALIDATION: Confirm every domain shows exactly one Data Owner, with no "shared" or blank entries.
ESCALATION: Domains with contested ownership route to executive sponsor for a definitive ruling.
```

Output: Explicit, single-owner data governance framework per domain.

Validate: Every domain confirmed to have exactly one named Data Owner.

Next: Contested ownership routes to executive sponsor for ruling.

LEVEL L2 HUMAN GATE Yes

DA-04 Data Quality Assessment

Use when: Assessing the actual quality of a dataset before it's relied on for a new use (e.g., reporting, AI).

Outcome: A data quality assessment scored across standard dimensions with evidence, not a general impression.

Inputs: Sample or full dataset, intended use case, [THRESHOLD] quality bar for that use

PROMPT

```
CONTEXT: Assessing data quality of <dataset> for intended use: <describe use case, e.g.,
regulatory reporting, AI training> on [PROJECT].
OBJECTIVE: Score data quality across Completeness, Accuracy, Consistency, Timeliness and
Uniqueness, with evidence, against the [THRESHOLD] bar required for the intended use.
INPUTS: Dataset (sample or full), profiling results if available, the intended use case's quality
requirements.
CONSTRAINTS: Quality bar must match the intended use — a bar sufficient for internal dashboards
may be insufficient for regulatory reporting or AI training; state the required bar explicitly.
DECISION RULES: Flag FIT FOR USE if all dimensions meet or exceed [THRESHOLD] for the intended
use; flag NOT FIT if any dimension falls short, naming the specific dimension and gap.
OUTPUT CONTRACT: Table [Dimension | Measured Score | Threshold Required | Gap | Evidence] +
overall Fit-for-Use verdict.
VALIDATION: Measured scores shown with the underlying evidence/method (e.g., % null values,
duplicate rate), not asserted.
ESCALATION: NOT FIT verdicts for regulatory or AI-training use route to the data owner for a
remediation plan before use proceeds.
```

Output: Dimension-scored data quality assessment with a use-case-matched fit verdict.

Validate: Measured scores shown with underlying evidence/method, not asserted.

Next: Not-fit datasets route to remediation before DA-08 AI/RAG readiness proceeds.

LEVEL L1 HUMAN GATE Conditional

DA-05 Data Lineage Mapping

Use when: You need to trace where a specific data element originates and where it flows, for audit or impact analysis.

Outcome: A traced lineage map from source system to consuming system(s), with transformation points documented.

Inputs: The data element/field in question, known system integrations, [SYSTEMS] involved

PROMPT

```
CONTEXT: Mapping data lineage for <data element/field> across [SYSTEMS] on [PROJECT].
OBJECTIVE: Trace the element from its system of record through every transformation to each
consuming system, documenting transformation logic at each hop.
INPUTS: The data element, known integration/ETL job documentation, system architecture.
CONSTRAINTS: Trace only through documented integration/transformation points — do not assume a
lineage path without evidence of the actual data flow.
DECISION RULES: Flag UNDOCUMENTED HOP if a transformation step is inferred from system behavior
but has no documentation — mark as INFERENCE, not confirmed lineage.
OUTPUT CONTRACT: Ordered lineage chain [Source | Transformation | Intermediate Store | Consuming
System] with each hop tagged CONFIRMED or INFERENCE.
VALIDATION: Confirm every CONFIRMED hop cites a specific integration/ETL job reference.
ESCALATION: Lineage gaps (INFERENCE hops) for regulated data route to DA-03 governance for formal
documentation.
```

Output: Source-to-consumer lineage chain with confirmed/inferred tagging per hop.

Validate: Every confirmed hop cites a specific integration/job reference.

Next: Inferred hops on regulated data route to governance for formal documentation.

LEVEL L2 HUMAN GATE No

DA-06 Data Migration Strategy

Use when: Migrating data between systems/platforms and needing a strategy for mapping, validation and cutover.

Outcome: A data migration strategy with explicit field mapping, validation approach and rollback plan.

Inputs: Source and target schema, data volume, [CONSTRAINTS] (downtime window)

PROMPT

```
CONTEXT: Planning data migration from <source> to <target> for [PROJECT], within [CONSTRAINTS].
OBJECTIVE: Define field-level mapping, transformation rules, validation approach and rollback
plan for the migration.
INPUTS: Source and target schema, data volume/complexity, allowed downtime window.
CONSTRAINTS: Every source field must have an explicit mapping decision (map, transform, or
explicitly drop with reason) — no field left unaccounted for.
DECISION RULES: Recommend a phased/incremental migration approach if data volume or downtime
constraints make a single cutover infeasible; otherwise single cutover with a validated rollback
point.
OUTPUT CONTRACT: Sections — Field Mapping Table (Source | Target | Transformation Rule | Notes) |
Validation Approach (reconciliation method) | Cutover Plan | Rollback Trigger and Procedure.
VALIDATION: Confirm every source field appears in the mapping table with an explicit disposition.
ESCALATION: Data loss risk on any dropped field routes to the business data owner for explicit
sign-off before migration.
```

Output: Field-complete migration strategy with validation and rollback plan.

Validate: Every source field confirmed present in the mapping table with a disposition.

Next: Dropped-field sign-off from data owner required before cutover.

LEVEL L2 HUMAN GATE Yes

DA-07 Analytics Architecture Design

Use when: Designing the architecture to support analytics/BI consumption at scale.

Outcome: An analytics architecture design matched to query patterns and freshness requirements, not a default stack choice.

Inputs: Analytics/BI consumption patterns, data freshness requirements, [CONSTRAINTS]

PROMPT

```
CONTEXT: Designing analytics architecture for <domain> on [PROJECT].
OBJECTIVE: Select architecture components (data warehouse, lakehouse, semantic layer, BI tool
integration) justified against actual query patterns and freshness needs.
INPUTS: Expected query complexity/volume, data freshness requirement (real-time vs daily batch),
existing tooling constraints.
CONSTRAINTS: Justify freshness-driving architecture choices (e.g., streaming ingestion) only
where the stated freshness requirement genuinely needs it — do not over-engineer for real-time
where daily batch suffices.
DECISION RULES: Recommend streaming/near-real-time ingestion only if the freshness requirement is
stated in minutes; otherwise recommend batch, which is simpler and cheaper.
OUTPUT CONTRACT: Sections — Consumption Pattern Summary | Architecture Components Selected with
Rationale | Freshness Design | Cost/Complexity Trade-off Note.
VALIDATION: Confirm the freshness design matches the stated requirement, not a default preference
for real-time.
ESCALATION: Cost/complexity trade-offs beyond current budget route to sponsor for a scope
decision.
```

Output: Freshness- and query-pattern-matched analytics architecture design.

Validate: Freshness design checked against the actual stated requirement.

Next: Escalate cost/complexity trade-offs beyond budget to sponsor.

LEVEL L2 HUMAN GATE No

DA-08 AI / RAG Data Readiness Assessment

Use when: Before building an AI/RAG solution, assessing whether the underlying data is actually ready to ground it.

Outcome: A data readiness assessment specific to AI/RAG use, flagging quality, structure and access gaps.

Inputs: Candidate data sources for grounding, [SUCCESS_CRITERIA] for the AI use case, current data quality (DA-04)

PROMPT

```
CONTEXT: Assessing data readiness for an AI/RAG solution on [PROJECT], grounding source(s): <name
sources>.
OBJECTIVE: Assess whether candidate data sources are ready to ground the AI use case reliably —
covering quality, structure, freshness and access control.
INPUTS: Candidate data sources, data quality assessment (DA-04) if available, access control
requirements for sensitive content.
CONSTRAINTS: Flag a source NOT READY if it contains unstructured content with no reliable
chunking/metadata strategy, or if access controls cannot be preserved through the retrieval layer
(risk of exposing restricted content).
DECISION RULES: READY if quality meets threshold (DA-04), structure/metadata supports reliable
retrieval, and access controls are preservable; else NOT READY with the specific blocking factor
named.
OUTPUT CONTRACT: Table [Data Source | Quality Status | Structure/Metadata Fit | Access Control
Preservable? | Overall Readiness | Blocking Factor (if not ready)].
VALIDATION: Confirm every NOT READY verdict names a specific blocking factor, not a general
"needs work."
ESCALATION: Access-control-preservation failures route to Security Architect before any AI use
case proceeds — this is a data leakage risk, not a quality nicety.
```

Output: AI/RAG-specific data readiness verdict per source with named blockers.

Validate: Not-ready verdicts confirmed to name a specific blocking factor.

Next: Access-control failures route to SEC review before AIX-06 RAG design proceeds.

LEVEL L2 HUMAN GATE Yes

Operations & Quality

28 Prompts Across 3 Practitioner Roles

I designed this section around a critical reality of technology delivery: **the quality of a solution is ultimately demonstrated through how reliably it can be built, tested, released, and operated.** These prompts help practitioners work with the evidence generated across that lifecycle—from pipeline behavior and reliability signals to test coverage, defects, release criteria, incidents, changes, and service performance.

Role	Prompts	Core Focus
DevOps / SRE / Platform Engineer	**10**	CI/CD, reliability, observability, incidents, capacity & platform automation
QA / Test Lead	**10**	Test strategy, traceability, risk-based testing, defects, automation & release quality
Service Delivery / IT Operations	**8**	Incidents, problems, changes, SLA performance & service improvement

Engineering for Reliability

The **DevOps / SRE / Platform** prompts focus on the operational characteristics that determine whether delivery can move safely into production. They cover areas such as pipeline design, deployment risk, SLO/SLA definition, error budgets, observability, reliability analysis, capacity, root-cause analysis, and opportunities for platform automation. The emphasis is on **measured signals and structural improvement**, rather than repeatedly treating operational symptoms.

Quality as Evidence

The **QA / Test Lead** prompts extend beyond generating test cases. They address traceability, risk-based prioritization, regression optimization, defect trends, automation opportunities, performance strategy, coverage gaps, and objective release-quality gates. This enables teams to connect requirements and risk to actual test evidence and make release decisions against **pre-defined criteria rather than subjective confidence**.

Service Outcomes

The **Service Delivery / IT Operations** prompts bring the perspective of the live service into the delivery lifecycle. Incident and problem analysis, change-risk assessment, and related operational practices help connect what happens in production back to engineering and delivery decisions. The objective is to turn operational experience into **repeatable learning and improvement**, rather than treating incidents as isolated events.

DevOps / SRE / Platform Engineer (OPS)

Pipeline design, reliability engineering and incident/RCA discipline grounded in measured signals.

OPS-01 CI/CD Pipeline Design Review

Use when: Designing or re-architecting a CI/CD pipeline for reliability and speed, not just patching a symptom.

Outcome: A pipeline design review addressing structural causes of instability/slowness, not stage-level symptoms.

Inputs: Current pipeline architecture, failure/duration history (or TL-10 output), [CONSTRAINTS]

PROMPT

```
CONTEXT: Redesigning the CI/CD pipeline for [SYSTEMS] on [PROJECT] following diagnosed
instability/slowness (see TL-10).
OBJECTIVE: Address the structural cause of pipeline issues (e.g., shared test environment
contention, lack of parallelization, flaky test architecture) rather than the symptomatic stage
alone.
INPUTS: Current pipeline architecture, stage-level failure/duration diagnostic, team constraints
(tooling, budget).
CONSTRAINTS: Distinguish a structural fix (e.g., test parallelization, environment isolation)
from a workaround (e.g., retry-on-failure) — prefer structural fixes for recurring issues.
DECISION RULES: Recommend structural redesign if the same stage has been flagged unstable/slow
across 3+ review cycles; recommend a targeted fix for a first-time occurrence.
OUTPUT CONTRACT: Sections — Root Structural Cause | Recommended Redesign | Migration/Rollout
Approach | Expected Reliability/Speed Improvement | Residual Risk.
VALIDATION: Confirm the recommendation targets the root structural cause identified, not a
superficial stage-level patch.
ESCALATION: Redesigns requiring new tooling/infrastructure investment route to engineering
leadership for budget approval.
```

Output: Structural pipeline redesign recommendation with expected improvement.

Validate: Recommendation confirmed to target the root structural cause, not a superficial patch.

Next: Budget-impacting redesigns route to engineering leadership.

LEVEL L2 HUMAN GATE Conditional

OPS-02 Deployment Risk Assessment

Use when: Before a significant deployment, assessing its specific risk profile beyond a generic checklist.

Outcome: A deployment risk assessment naming the specific blast radius and rollback trigger, not a generic go/no-go.

Inputs: Deployment scope/change description, [SYSTEMS] affected, rollback capability

PROMPT

```
CONTEXT: Assessing deployment risk for <release/change> on [PROJECT] affecting [SYSTEMS].
OBJECTIVE: Identify the specific blast radius if the deployment fails, and confirm a tested
rollback path exists.
INPUTS: Change description/diff scope, dependent systems/consumers, current rollback mechanism
and last test date.
CONSTRAINTS: Blast radius must be described specifically (which systems/users/data are affected
if this fails) — not a generic "could impact production."
DECISION RULES: Flag HIGH RISK if the rollback mechanism is untested within a reasonable recency
window, or if the change touches a shared/critical-path component with no canary/staged rollout
plan.
OUTPUT CONTRACT: Sections — Blast Radius (specific) | Rollback Mechanism & Last Tested | Staged
Rollout Plan (canary %, monitoring gates) | Risk Rating | Go/No-Go Recommendation.
VALIDATION: Confirm the blast radius statement names specific affected systems/users, not a
generic statement.
ESCALATION: HIGH RISK deployments require named sign-off before proceeding, per change management
policy.
```

Output: Specific blast-radius deployment risk assessment with a go/no-go recommendation.

Validate: Blast radius statement checked for naming specific affected systems/users.

Next: High-risk deployments require named sign-off per change policy.

LEVEL L1 HUMAN GATE Yes

OPS-03 Incident Response Analysis

Use when: During or immediately after an incident, structuring the response and initial findings.

Outcome: A structured incident timeline and initial impact assessment, ready for stakeholder communication.

Inputs: Incident timeline/logs, affected [SYSTEMS], [AUDIENCE] for communication

PROMPT

```
CONTEXT: Analyzing incident <ID/description> affecting [SYSTEMS] on [PROJECT], communicating to
[AUDIENCE].
OBJECTIVE: Produce a structured timeline and impact assessment to support response coordination
and stakeholder communication.
INPUTS: Incident timeline/logs/alerts, affected systems and user impact data, current mitigation
status.
CONSTRAINTS: State impact in concrete, measurable terms (users affected, transactions failed,
duration) — not "significant impact."
DECISION RULES: Classify severity by actual measured impact (users/revenue/data affected) against
the organization's defined severity matrix, not by initial alarm level alone.
OUTPUT CONTRACT: Sections — Timeline (time-stamped) | Current Status | Measured Impact |
Mitigation Actions Taken | Next Update Due.
VALIDATION: Confirm impact figures are measured/estimated from actual data (monitoring, logs),
not assumed.
ESCALATION: This is a live-incident artifact — route immediately per incident command protocol;
do not hold for review.
```

Output: Time-stamped incident status and impact summary for live communication.

Validate: Impact figures confirmed measured from actual monitoring/log data.

Next: Feed into OPS-09 Root Cause Analysis once mitigated.

LEVEL L1 HUMAN GATE No

OPS-04 SLO/SLA Definition

Use when: Defining or refreshing service level objectives/agreements for a service.

Outcome: SLOs defined from actual user-experience-relevant signals, with SLA derived conservatively from SLO.

Inputs: Service criticality, [AUDIENCE]/consumer expectations, current performance baseline

PROMPT

```
CONTEXT: Defining SLO/SLA for <service> on [PROJECT], consumed by [AUDIENCE].
OBJECTIVE: Define SLOs on signals that reflect actual user experience, and derive a conservative
SLA from measured SLO performance.
INPUTS: Service criticality, consumer expectations/contractual context, current measured
performance baseline.
CONSTRAINTS: SLO targets must be achievable based on current measured baseline plus a stated
improvement margin — do not set an aspirational target disconnected from current performance
without an improvement plan.
DECISION RULES: Set SLA (external commitment) below the internal SLO target, preserving an error-
budget buffer for internal improvement work.
OUTPUT CONTRACT: Table [Signal (e.g., availability, latency p99) | Current Baseline | SLO Target
| SLA Commitment | Error Budget Buffer] + measurement method per signal.
VALIDATION: Confirm SLA is set below SLO with an explicit buffer shown, not equal to it.
ESCALATION: SLA commitments with contractual/financial penalties route to legal/commercial review
before publication.
```

Output: Baseline-grounded SLO/SLA definition with an explicit error-budget buffer.

Validate: SLA confirmed set below SLO with an explicit buffer shown.

Next: Legal/commercial review for penalty-bearing SLAs before publication.

LEVEL L1 HUMAN GATE Yes

OPS-05 Error Budget Analysis

Use when: Determining whether the team has room to ship risky changes or must prioritize reliability work.

Outcome: An error budget analysis with a clear verdict on whether the team can ship or must focus on reliability.

Inputs: SLO target, actual measured performance over the period, planned risky changes

PROMPT

```
CONTEXT: Analyzing error budget status for <service> on [PROJECT] against its SLO.
OBJECTIVE: Determine remaining error budget for the period and whether it supports planned risky
changes or requires a reliability freeze.
INPUTS: SLO target, actual measured performance/incidents over the period, list of planned
changes with risk level.
CONSTRAINTS: Calculate remaining budget from actual measured performance against the SLO target —
do not estimate qualitatively.
DECISION RULES: If remaining budget is exhausted or trending to exhaustion before period end,
recommend a feature freeze / reliability focus; if budget is healthy, approve planned risky
changes with monitoring conditions.
OUTPUT CONTRACT: Sections — SLO Target | Actual Performance | Budget Consumed % | Budget
Remaining | Recommendation (Freeze/Proceed with conditions) | Trend.
VALIDATION: Budget consumed percentage recalculated from actual measured performance vs SLO
target.
ESCALATION: Recommended freezes affecting committed roadmap dates route to TDM/PM for delivery-
plan adjustment.
```

Output: Calculated error budget status with a ship/freeze recommendation.

Validate: Budget-consumed percentage recalculated from actual performance data.

Next: Freeze recommendations route to TDM/PM for delivery-plan adjustment.

LEVEL L1 HUMAN GATE Conditional

OPS-06 Observability Strategy (Platform-Level)

Use when: Implementing platform-wide observability standards across multiple services/teams.

Outcome: A platform observability standard defining required signal types and instrumentation baseline per service.

Inputs: Service inventory, current instrumentation state, [CONSTRAINTS] (tooling)

PROMPT

```
CONTEXT: Defining platform-wide observability standards for [PROGRAM] across multiple
services/teams.
OBJECTIVE: Define a minimum instrumentation baseline (logs, metrics, traces) every service must
meet, and assess current gap against it.
INPUTS: Service inventory, current instrumentation coverage per service, available
tooling/platform constraints.
CONSTRAINTS: The baseline must specify concrete requirements (e.g., "structured logs with
correlation ID," "RED metrics exposed," "distributed tracing on cross-service calls") — not
"adequate logging."
DECISION RULES: Flag NON-COMPLIANT if a service lacks any baseline element; prioritize
remediation for services with the highest incident frequency or criticality first.
OUTPUT CONTRACT: Baseline definition (list of required signals) + compliance table [Service |
Compliant Elements | Gaps | Priority].
VALIDATION: Confirm the baseline definition is concrete and specific enough to be objectively
checked, not aspirational.
ESCALATION: Critical services with major gaps route to team leads for a remediation sprint
commitment.
```

Output: Concrete platform observability baseline with per-service compliance gaps.

Validate: Baseline definition confirmed concrete and objectively checkable.

Next: Critical gaps feed a remediation sprint commitment per team.

LEVEL L2 HUMAN GATE No

OPS-07 Reliability Engineering Review

Use when: A broader review of a service's reliability engineering practices, beyond a single incident.

Outcome: A reliability practice review identifying systemic gaps (testing, chaos engineering, capacity, on-call).

Inputs: Current reliability practices, incident history trend, [SUCCESS_CRITERIA]

PROMPT

```
CONTEXT: Reviewing reliability engineering practices for <service/team> on [PROJECT].
OBJECTIVE: Identify systemic practice gaps (failure testing, capacity planning, on-call health,
deployment safety) contributing to reliability trend.
INPUTS: Current practices in each area, incident history/trend over recent periods, target
reliability in [SUCCESS_CRITERIA].
CONSTRAINTS: Connect a practice gap to the incident trend only where evidence supports it (e.g.,
recurring incident category with no corresponding test coverage) — do not assume all gaps are
equally causal.
DECISION RULES: Prioritize gaps that correlate with the most frequent or highest-severity
incident category in the trend data.
OUTPUT CONTRACT: Table [Practice Area | Current State | Gap | Linked Incident Pattern (if
evidenced) | Priority | Recommendation].
VALIDATION: Confirm linked-incident-pattern claims are backed by an actual incident category
match, not assumption.
ESCALATION: On-call health gaps indicating burnout risk route to engineering leadership for
staffing review — a people, not tooling, escalation.
```

Output: Evidence-linked reliability practice gap review with prioritized recommendations.

Validate: Linked-incident claims confirmed backed by an actual category match.

Next: On-call burnout risk routes to engineering leadership for staffing review.

LEVEL L2 HUMAN GATE Conditional

OPS-08 Capacity & Scaling Analysis

Use when: Assessing whether current auto-scaling/capacity configuration actually matches real traffic patterns.

Outcome: A scaling configuration analysis identifying mismatches between scaling policy and actual traffic behavior.

Inputs: Current scaling policy/thresholds, actual traffic pattern data, incident history related to scaling

PROMPT

```
CONTEXT: Analyzing scaling configuration for <service> on [PROJECT] against actual traffic
patterns.
OBJECTIVE: Identify mismatches between current scaling policy (thresholds, cooldowns, min/max)
and actual observed traffic behavior.
INPUTS: Current auto-scaling configuration, traffic pattern data (including spikes/seasonality),
any scaling-related incidents.
CONSTRAINTS: Evaluate scaling responsiveness against actual measured spike shape (how fast
traffic rises) — a scaling policy tuned for gradual growth will fail for sharp spikes regardless
of max-capacity settings.
DECISION RULES: Flag POLICY MISMATCH if the scale-up cooldown/threshold is slower than the
observed typical time-to-spike, causing under-provisioning during ramp.
OUTPUT CONTRACT: Table [Metric | Current Policy Setting | Observed Traffic Behavior | Mismatch? |
Recommended Setting].
VALIDATION: Confirm the mismatch flag is checked against actual observed spike-shape data, not
generic best practice.
ESCALATION: Mismatches that caused past customer-facing incidents route to immediate policy
update, not the next planning cycle.
```

Output: Traffic-pattern-matched scaling policy analysis with specific mismatches.

Validate: Mismatch flags checked against actual observed spike-shape data.

Next: Incident-linked mismatches get immediate policy updates.

LEVEL L1 HUMAN GATE No

OPS-09 Root Cause Analysis (RCA)

Use when: After an incident, producing a rigorous RCA that distinguishes root cause from contributing factors.

Outcome: A rigorous RCA using a structured method (e.g., 5 Whys / fishbone) distinguishing root cause from symptoms.

Inputs: Incident timeline (OPS-03), logs/evidence, contributing systems/changes

PROMPT

```
CONTEXT: Conducting RCA for incident <ID> affecting [SYSTEMS] on [PROJECT].
OBJECTIVE: Identify the true root cause (the earliest point where a different action would have
prevented the incident), distinct from contributing factors and symptoms.
INPUTS: Incident timeline, logs/traces, recent changes/deployments, on-call response notes.
CONSTRAINTS: Distinguish FACT (confirmed in logs/data), INFERENCE (reasonable read of evidence)
and ASSUMPTION (unverified) for every causal claim.
DECISION RULES: A cause is ROOT CAUSE only if addressing it would have prevented this incident
category from occurring at all; contributing factors made it worse or slower to detect/resolve
but wouldn't alone have prevented it.
OUTPUT CONTRACT: Sections — Timeline Summary | Root Cause (with Fact/Inference/Assumption
tagging) | Contributing Factors | Detection Gap (how long to detect, why) | Corrective Actions
(each with owner and date) | Preventive Actions.
VALIDATION: Root cause statement backed by at least one FACT-tagged evidence point.
ESCALATION: Corrective actions requiring cross-team investment route to engineering leadership
for prioritization against other work.
```

Output: Structured RCA with fact/inference/assumption-tagged root cause and action plan.

Validate: Root cause statement backed by at least one FACT-tagged evidence point.

Next: Track corrective/preventive actions to completion; review at next reliability cycle.

LEVEL L2 HUMAN GATE No

OPS-10 Platform Automation Candidate Assessment

Use when: Identifying which manual operational tasks are worth automating, ranked by actual ROI.

Outcome: A ranked automation candidate list based on frequency, toil and error cost, not novelty.

Inputs: List of manual operational tasks/runbooks, frequency data, time cost per execution

PROMPT

```
CONTEXT: Assessing automation candidates among manual operational tasks for [SYSTEMS] on
[PROJECT].
OBJECTIVE: Rank candidates by expected ROI (frequency × time saved − automation build cost) and
human-error risk reduced.
INPUTS: Task/runbook list, execution frequency, time cost per execution, historical error rate
from manual execution if known.
CONSTRAINTS: Estimate automation build cost realistically (including maintenance) — do not
present automation as free once built.
DECISION RULES: Prioritize HIGH if (frequency × time saved per year) recoups estimated
build+maintenance cost within a reasonable payback window (e.g., under 6 months); otherwise
MEDIUM/LOW.
OUTPUT CONTRACT: Table [Task | Frequency | Time Cost/Execution | Annual Time Cost | Est.
Automation Cost | Payback Period | Error Reduction Potential | Priority].
VALIDATION: Payback period recalculated from the stated frequency, time cost and automation cost
figures.
ESCALATION: High-error-rate manual tasks touching production data route to priority regardless of
payback period, given risk reduction value.
```

Output: ROI-ranked automation candidate list with calculated payback periods.

Validate: Payback period recalculated from the stated frequency/cost figures.

Next: High-priority candidates feed the platform engineering backlog.

LEVEL L1 HUMAN GATE No

QA / Test Lead (QA)

Test strategy, traceability and release quality decisions grounded in risk and evidence.

QA-01 Test Strategy Design

Use when: Defining the overall test approach for a project or release, beyond a generic test plan template.

Outcome: A risk-weighted test strategy allocating effort where failure consequence is highest.

Inputs: [PROJECT] scope, risk profile, [CONSTRAINTS] (timeline, environments)

PROMPT

```
CONTEXT: Designing the test strategy for [PROJECT] within [CONSTRAINTS].
OBJECTIVE: Allocate test effort/technique (unit, integration, E2E, performance, security,
exploratory) proportional to risk and failure consequence, not evenly.
INPUTS: Scope/architecture, known high-risk areas (new tech, complex logic, regulatory-sensitive
flows), timeline/environment constraints.
CONSTRAINTS: Justify test technique allocation against the specific risk profile of each area —
do not apply a uniform testing approach across low- and high-risk components.
DECISION RULES: Allocate the deepest test coverage (multiple techniques, higher automation
investment) to areas with both high change frequency and high failure consequence.
OUTPUT CONTRACT: Table [Area/Component | Risk Level | Test Techniques Applied | Automation Level
| Rationale] + overall test approach narrative and environment/data needs.
VALIDATION: Confirm the highest-risk areas show the most rigorous technique combination, not a
uniform default.
ESCALATION: Areas with high risk but insufficient timeline for adequate coverage route to PM-
06/TDM-04 risk escalation.
```

Output: Risk-weighted test strategy with technique allocation rationale per area.

Validate: Highest-risk areas confirmed to show the most rigorous technique combination.

Next: Insufficient-coverage risk routes to TDM-04/PM-05 for risk treatment.

LEVEL L2 HUMAN GATE No

QA-02 Test Plan Generator

Use when: Converting acceptance criteria/requirements into an executable test plan for a specific feature/release.

Outcome: A test plan with concrete test cases traced to requirements, covering happy path and key negative cases.

Inputs: Acceptance criteria (BA-05), requirement set, [CONSTRAINTS] (environments/data)

PROMPT

```
CONTEXT: Generating the test plan for <feature/release> on [PROJECT] from defined acceptance
criteria.
OBJECTIVE: Produce concrete, executable test cases tracing to each acceptance criterion, covering
positive, negative and boundary conditions.
INPUTS: Acceptance criteria set, requirement context, environment/test data constraints.
CONSTRAINTS: Every acceptance criterion must map to at least one test case — no criterion left
uncovered without an explicit DATA GAP note.
DECISION RULES: Include a negative/boundary test case wherever the acceptance criterion implies a
validation rule or limit (e.g., input length, numeric range).
OUTPUT CONTRACT: Table [Test Case ID | Linked Criterion | Test Type (Positive/Negative/Boundary)
| Steps | Expected Result | Test Data Needed].
VALIDATION: Confirm every acceptance criterion has at least one linked test case; flag any
without.
ESCALATION: Criteria with no feasible test data/environment route to environment/data
provisioning request before execution.
```

Output: Criterion-traced, executable test case set covering key condition types.

Validate: Every acceptance criterion confirmed to have at least one linked test case.

Next: Feed into QA-03 Traceability and execution scheduling.

LEVEL L1 HUMAN GATE No

QA-03 Requirements-to-Test Traceability

Use when: Confirming complete test coverage against requirements for audit or release-readiness purposes.

Outcome: A traceability check confirming every requirement has test coverage, with orphans and untested items flagged.

Inputs: Requirements list, test case inventory, execution results if available

PROMPT

```
CONTEXT: Verifying requirements-to-test traceability for [PROJECT] release <name/version>.
OBJECTIVE: Confirm every requirement has documented test coverage and every test case links back
to a requirement; flag orphans in either direction.
INPUTS: Requirements list, test case inventory with linked requirement IDs, execution status if
available.
CONSTRAINTS: A link is valid only if explicitly documented in the test case metadata — do not
infer coverage from similar naming or topic.
DECISION RULES: Flag UNTESTED if a requirement has zero linked test cases. Flag ORPHAN TEST if a
test case links to no valid requirement (may indicate scope drift).
OUTPUT CONTRACT: Table [Requirement ID | Linked Test Cases | Execution Status | Coverage Status
(Covered/Untested)] + separate orphan test case list.
VALIDATION: Spot-check 3 "Covered" requirements to confirm the linked test cases explicitly
reference them.
ESCALATION: UNTESTED requirements on release-blocking scope route to SM-10 Release Readiness
Assessment as a blocking finding.
```

Output: Requirement-and-test bidirectional traceability check with orphan flags.

Validate: Sampled covered requirements confirmed to have explicit test case links.

Next: Untested release-blocking items feed SM-10 Release Readiness.

LEVEL L1 HUMAN GATE No

QA-04 Risk-Based Test Prioritization

Use when: Test execution time is constrained and test cases need prioritization by actual risk.

Outcome: A prioritized test execution order maximizing risk coverage within the available time window.

Inputs: Full test case inventory, [CONSTRAINTS] (available execution time), area risk ratings

PROMPT

```
CONTEXT: Prioritizing test execution for [PROJECT] release within a constrained execution window:
[CONSTRAINTS].
OBJECTIVE: Order test execution to maximize risk coverage first, given the time available may not
permit full execution.
INPUTS: Test case inventory with linked component/area, area risk ratings (from QA-01 or change
impact data), available execution time.
CONSTRAINTS: Prioritize by (failure consequence × likelihood of a defect in that area, e.g.,
recent change volume) — do not prioritize by convenience or test-writer availability.
DECISION RULES: Tier 1 (must-run) = high risk + recently changed areas; Tier 2 = high risk +
stable areas or medium risk + recently changed; Tier 3 = low risk, stable areas, run only if time
permits.
OUTPUT CONTRACT: Tiered table [Tier | Test Cases | Area | Risk Basis | Est. Execution Time] +
cumulative time vs available window.
VALIDATION: Confirm Tier 1 classification checked against both risk and recent-change criteria,
not either alone.
ESCALATION: If Tier 1 alone exceeds the available window, escalate the timeline/scope conflict to
PM/TDM before proceeding.
```

Output: Risk-tiered test execution plan fit to the available time window.

Validate: Tier 1 classification checked against both risk and recent-change criteria.

Next: Timeline conflicts escalate to PM-06/TDM-08 for a resourcing decision.

LEVEL L1 HUMAN GATE Conditional

QA-05 Regression Suite Optimization

Use when: The regression suite has grown bloated/slow and needs pruning based on actual defect-catching value.

Outcome: An optimized regression suite retaining high-value tests and flagging low-value/redundant ones for removal.

Inputs: Regression suite inventory, historical defect-catch data per test, execution time per test

PROMPT

```
CONTEXT: Optimizing the regression suite for [SYSTEMS] on [PROJECT], currently taking <duration>
to execute.
OBJECTIVE: Identify low-value/redundant tests for removal or consolidation, based on actual
historical defect-catching data, not test count reduction targets alone.
INPUTS: Regression suite inventory, historical pass/fail and defect-catch record per test,
execution time per test.
CONSTRAINTS: A test qualifies as LOW VALUE only if it has caught zero real defects across a
meaningful historical window AND its coverage significantly overlaps another test — do not remove
tests solely for being slow.
DECISION RULES: Flag REMOVE CANDIDATE if zero defect catches in the window and >80% functional
overlap with another retained test. Flag KEEP-CRITICAL for tests covering regulatory/high-
severity-incident-linked scenarios regardless of catch rate.
OUTPUT CONTRACT: Table [Test | Defect Catches (period) | Execution Time | Overlap With |
Recommendation] + estimated suite time savings.
VALIDATION: Confirm REMOVE CANDIDATE entries show both zero catches and a named overlapping test.
ESCALATION: KEEP-CRITICAL exclusions from removal are non-negotiable regardless of suite
optimization pressure.
```

Output: Evidence-based regression suite optimization with quantified time savings.

Validate: Remove candidates confirmed to show both zero catches and a named overlap.

Next: Implement removals incrementally; monitor defect escape rate post-change.

LEVEL L2 HUMAN GATE Yes

QA-06 Defect Trend Analysis

Use when: You need to understand whether defect patterns indicate a systemic quality issue, not just count trends.

Outcome: A defect trend analysis identifying the specific component/cause category driving the trend.

Inputs: Defect log with component/category/severity tags, time period for comparison

PROMPT

```
CONTEXT: Analyzing defect trends for [SYSTEMS] on [PROJECT] over <period> vs prior period.
OBJECTIVE: Identify whether rising/falling defect counts reflect a systemic issue in a specific
component or cause category, not just overall volume noise.
INPUTS: Defect log with component, root cause category and severity tags across both periods.
CONSTRAINTS: Break down the trend by component and cause category before drawing conclusions — a
rising overall count may be driven by one component while others improve.
DECISION RULES: Flag SYSTEMIC CONCERN if a single component or cause category accounts for a
disproportionate share (e.g., >40%) of the period's defects or shows a worsening trend across 2+
periods.
OUTPUT CONTRACT: Table [Component/Category | This Period | Prior Period | Trend | % of Total] +
narrative naming the primary driver of any overall trend.
VALIDATION: Confirm the stated "primary driver" is the component/category with the largest share
shown in the table, not a general impression.
ESCALATION: Systemic concerns tied to a specific engineering area route to TL-03 Technical Debt
Assessment for that component.
```

Output: Component-level defect trend breakdown with a named primary driver.

Validate: Stated primary driver checked against the actual largest-share component in the data.

Next: Systemic component concerns route to TL-03 Technical Debt Assessment.

LEVEL L1 HUMAN GATE No

QA-07 Test Automation Candidate Assessment

Use when: Deciding which manual test cases are worth automating, ranked by actual ROI.

Outcome: A ranked automation candidate list weighing execution frequency, stability and maintenance cost.

Inputs: Manual test case inventory, execution frequency, historical stability of the feature under test

PROMPT

```
CONTEXT: Assessing test automation candidates for [SYSTEMS] on [PROJECT].
OBJECTIVE: Rank manual test cases for automation by ROI (execution frequency × manual time cost,
offset by expected automation maintenance burden).
INPUTS: Test case inventory, execution frequency (per release/sprint), feature stability (churn
rate — automating an unstable UI/flow yields high maintenance cost).
CONSTRAINTS: Do not recommend automating high-churn, unstable features as a first priority —
maintenance cost will likely exceed the benefit; flag these as LOW PRIORITY regardless of
frequency.
DECISION RULES: HIGH priority = high execution frequency + stable feature; LOW priority = any
feature with high UI/logic churn in recent periods, regardless of frequency.
OUTPUT CONTRACT: Table [Test Case | Execution Frequency | Manual Time Cost | Feature Stability |
Priority | Rationale].
VALIDATION: Confirm LOW priority classification for unstable features is applied regardless of
frequency, per the decision rule.
ESCALATION: N/A — feed directly into automation backlog prioritization.
```

Output: ROI- and stability-weighted test automation candidate ranking.

Validate: Low-priority classification confirmed applied to unstable features regardless of frequency.

Next: Feed HIGH priority items into the automation engineering backlog.

LEVEL L1 HUMAN GATE No

QA-08 Performance Test Strategy

Use when: Designing a performance test approach tied to real usage patterns and defined performance targets.

Outcome: A performance test strategy with load profiles derived from real usage data, not invented numbers.

Inputs: Real/projected usage data, target [SUCCESS_CRITERIA] (latency/throughput), [SYSTEMS] under test

PROMPT

```
CONTEXT: Designing performance test strategy for [SYSTEMS] on [PROJECT] against target
[SUCCESS_CRITERIA].
OBJECTIVE: Define load profiles (normal, peak, stress, soak) derived from actual/projected usage
data, and the specific pass/fail thresholds per profile.
INPUTS: Real usage/traffic data or projection basis, target latency/throughput/error-rate
criteria.
CONSTRAINTS: Load profile numbers must be derived from stated usage data or a documented growth
projection — do not invent round numbers.
DECISION RULES: Include a Soak test (sustained load over extended duration) if the system
maintains state/connections over time (memory leak/connection exhaustion risk); otherwise short-
duration load/stress tests may suffice.
OUTPUT CONTRACT: Table [Profile (Normal/Peak/Stress/Soak) | Load Level (with derivation basis) |
Duration | Pass/Fail Threshold] + test environment representativeness note.
VALIDATION: Confirm each load level cites its derivation from actual usage/projection data.
ESCALATION: Test environment materially different from production (sizing, data volume) routes to
a caveat in the results — do not present results as production-equivalent.
```

Output: Usage-derived performance test strategy with explicit pass/fail thresholds.

Validate: Each load level checked for a cited derivation from usage/projection data.

Next: Execute; feed results into TL-08 Performance Bottleneck Analysis if thresholds fail.

LEVEL L2 HUMAN GATE No

QA-09 Release Quality Gate Assessment

Use when: Formal go/no-go quality gate before a release, requiring an evidence-based verdict.

Outcome: A quality gate verdict against pre-defined, objective criteria, not subjective team confidence.

Inputs: Pre-defined quality gate criteria, current test results/defect status, [THRESHOLD]

PROMPT

```
CONTEXT: Assessing the quality gate for <release> on [PROJECT] against pre-defined criteria.
OBJECTIVE: Issue a pass/fail quality gate verdict strictly against pre-defined, objective
criteria.
INPUTS: Quality gate criteria (e.g., test pass rate, open defect severity counts, coverage %),
current actual results.
CONSTRAINTS: Evaluate strictly against the pre-defined criteria — do not adjust the bar post-hoc
to accommodate current results (this is the same anti-bias discipline as SA-11 POC Evaluation).
DECISION RULES: PASS only if every criterion meets its threshold; CONDITIONAL PASS if only non-
critical criteria are short, with named accepted risk; FAIL if any critical criterion (e.g., zero
Critical defects) is not met.
OUTPUT CONTRACT: Table [Criterion | Threshold | Actual | Met? (Y/N)] + overall verdict + accepted
risks if Conditional.
VALIDATION: Confirm the criteria list matches the originally defined gate, with no post-hoc
changes.
ESCALATION: FAIL or CONDITIONAL PASS verdicts require named release-owner sign-off before
proceeding.
```

Output: Objective quality gate verdict against pre-defined, unaltered criteria.

Validate: Criteria list confirmed unchanged from the original gate definition.

Next: Route verdict to SM-10 Release Readiness Assessment for the final go/no-go.

LEVEL L1 HUMAN GATE Yes

QA-10 Test Coverage Gap Analysis

Use when: Assessing where test coverage is genuinely thin, beyond a raw code-coverage percentage.

Outcome: A coverage gap analysis identifying specific untested paths/scenarios, not just a coverage percentage.

Inputs: Code/functional coverage data, [SYSTEMS] critical paths, recent defect escape data

PROMPT

```
CONTEXT: Analyzing test coverage gaps for [SYSTEMS] on [PROJECT].
OBJECTIVE: Identify specific untested critical paths and scenarios, using coverage data plus
defect escape evidence — not a bare coverage percentage.
INPUTS: Code/functional coverage reports, list of business-critical paths, recent production
defect escapes (bugs that reached production despite testing).
CONSTRAINTS: A high overall coverage percentage does not mean adequate coverage — cross-reference
against critical paths and recent escapes specifically.
DECISION RULES: Flag CRITICAL GAP if a business-critical path shows low coverage or if a recent
production defect escape traces to an untested scenario.
OUTPUT CONTRACT: Table [Critical Path/Scenario | Coverage Status | Recent Escape Linked? | Gap
Severity | Recommendation].
VALIDATION: Confirm "linked escape" entries trace to an actual recent defect record, not
assumption.
ESCALATION: Critical gaps on regulatory or financial paths route to QA-02 Test Plan Generator for
immediate test case creation.
```

Output: Critical-path-focused coverage gap analysis linked to actual defect escapes.

Validate: Linked-escape entries confirmed traced to an actual defect record.

Next: Critical gaps route to QA-02 for immediate test case creation.

LEVEL L1 HUMAN GATE No

Service Delivery / IT Operations (SVC)

ITSM-aligned incident, problem, change and SLA management grounded in measured service data.

SVC-01 Incident Analysis

Use when: An incident needs structured analysis for classification, impact and initial response coordination.

Outcome: A classified incident analysis with measured impact, ready for response coordination and communication.

Inputs: Incident details/logs, affected service, [AUDIENCE] for communication

PROMPT

```
CONTEXT: Analyzing incident affecting <service> for [CUSTOMER], to be communicated to [AUDIENCE].
OBJECTIVE: Classify severity from measured impact and produce a response-ready summary.
INPUTS: Incident logs/monitoring data, affected user/transaction count, service criticality tier.
CONSTRAINTS: Classify severity using the organization's defined severity matrix against measured
impact — not the reporter's initial alarm level.
DECISION RULES: Severity = function of (users/transactions affected × service criticality tier ×
data sensitivity if applicable), per the defined matrix.
OUTPUT CONTRACT: Sections — Severity (with matrix justification) | Measured Impact | Current
Status | Next Update Cadence | Communication Draft for [AUDIENCE].
VALIDATION: Confirm severity classification traces to the specific matrix criteria and measured
figures, not initial impression.
ESCALATION: This is a live-incident artifact — route per incident management protocol
immediately.
```

Output: Matrix-justified incident severity classification with a ready communication draft.

Validate: Severity classification traced to specific matrix criteria and measured figures.

Next: Feed into SVC-02 Problem Management once stabilized.

LEVEL L1 HUMAN GATE No

SVC-02 Problem Management (RCA)

Use when: After incident closure, determining the underlying problem to prevent recurrence.

Outcome: A problem record with root cause distinguished from workaround, and a permanent fix tracked separately.

Inputs: Closed incident(s), workaround applied, [SOURCE_DATA] logs/evidence

PROMPT

```
CONTEXT: Conducting problem management analysis following incident(s) affecting <service> for
[CUSTOMER].
OBJECTIVE: Identify the underlying root cause distinct from the workaround applied during the
incident, and define a permanent fix.
INPUTS: Incident record(s), workaround details, supporting logs/evidence.
CONSTRAINTS: Distinguish WORKAROUND (restored service, did not fix the cause) from PERMANENT FIX
(addresses root cause) explicitly — do not close the problem record on a workaround alone.
DECISION RULES: Problem record stays OPEN until a permanent fix is implemented and verified, even
if all related incidents are closed.
OUTPUT CONTRACT: Sections — Root Cause (Fact/Inference/Assumption tagged) | Workaround Applied |
Permanent Fix Plan (owner, date) | Related Incidents | Status.
VALIDATION: Root cause statement backed by at least one FACT-tagged evidence point.
ESCALATION: Recurring incidents linked to the same open problem route to service delivery
leadership for priority escalation.
```

Output: Problem record separating workaround from permanent fix, with tracked status.

Validate: Root cause statement backed by at least one FACT-tagged evidence point.

Next: Track to permanent fix completion; verify no recurrence.

LEVEL L2 HUMAN GATE No

SVC-03 Change Risk Assessment

Use when: A change request needs risk assessment before change advisory board approval.

Outcome: A change risk assessment with specific failure impact and rollback confirmation, not a generic risk score.

Inputs: Change description, affected [SYSTEMS], rollback plan, change window

PROMPT

```
CONTEXT: Assessing risk for change request: <describe change> affecting <service/system> for
[CUSTOMER].
OBJECTIVE: Determine specific failure impact if the change goes wrong, and confirm rollback
readiness, for CAB decision.
INPUTS: Change description/scope, affected systems and dependent services, rollback plan and last
test date, proposed change window.
CONSTRAINTS: State failure impact specifically (which services/users affected, for how long) —
not a generic risk category label alone.
DECISION RULES: Flag HIGH RISK if rollback is untested or unavailable, or if the change window
overlaps a business-critical period with no compensating control.
OUTPUT CONTRACT: Sections — Change Summary | Specific Failure Impact if Unsuccessful | Rollback
Readiness | Recommended Risk Rating | CAB Recommendation.
VALIDATION: Confirm the failure impact statement names specific affected services/users, not a
generic label.
ESCALATION: HIGH RISK changes require CAB review before implementation — not an
emergency/standard change classification bypass.
```

Output: Specific-impact change risk assessment with a CAB-ready recommendation.

Validate: Failure impact statement checked for naming specific affected services/users.

Next: Route to CAB for formal approval per risk rating.

LEVEL L1 HUMAN GATE Yes

SVC-04 SLA Performance Review

Use when: Reviewing actual SLA performance against contracted commitments for a reporting period.

Outcome: An SLA performance review with measured figures against each contracted metric, trend included.

Inputs: Contracted SLA metrics/thresholds, measured performance data for the period

PROMPT

```
CONTEXT: Reviewing SLA performance for <service> for [CUSTOMER], period ending [DATE].
OBJECTIVE: Report measured performance against each contracted SLA metric, with trend and breach
implications.
INPUTS: Contracted SLA metrics and thresholds, measured performance data for the period and prior
period.
CONSTRAINTS: Report every contracted metric, not just the favorable ones — omission of an
unfavorable metric is not acceptable.
DECISION RULES: Flag BREACH if measured performance falls below the contracted threshold; flag AT
RISK if trending toward breach based on the current-vs-prior trend.
OUTPUT CONTRACT: Table [SLA Metric | Threshold | Measured (This Period) | Measured (Prior Period)
| Status | Trend] + breach implications if contractual penalties apply.
VALIDATION: Confirm every contracted metric appears in the table, with none omitted.
ESCALATION: BREACH results with contractual penalty implications route to commercial/account
management for customer communication.
```

Output: Complete, trend-inclusive SLA performance review against every contracted metric.

Validate: Every contracted metric confirmed present in the report table.

Next: Breach items route to commercial/account management for customer communication.

LEVEL L1 HUMAN GATE Conditional

SVC-05 Service Health Dashboard Narrative

Use when: Translating raw service health dashboard data into a narrative for stakeholders who don't read dashboards directly.

Outcome: A narrative summary of service health highlighting what changed and what needs attention.

Inputs: Dashboard/monitoring data snapshot, [AUDIENCE], [DATE]

PROMPT

```
CONTEXT: Producing the service health narrative for <service(s)> for [AUDIENCE], as of [DATE].
OBJECTIVE: Translate dashboard data into a narrative focused on what changed and what needs
attention, not a re-listing of every metric.
INPUTS: Current dashboard/monitoring snapshot, prior period snapshot for comparison.
CONSTRAINTS: Report deltas and exceptions prominently; stable/unchanged metrics get a brief roll-
up mention, not equal narrative weight.
DECISION RULES: Lead with any metric that moved from within-threshold to breach, or any new trend
line; deprioritize unchanged healthy metrics.
OUTPUT CONTRACT: Sections — Headline | What Changed | Needs Attention (with owner) | Stable
Metrics Roll-up (brief) | Next Review Date.
VALIDATION: Confirm "What Changed" items are genuine deltas versus the prior snapshot, not
restated current state.
ESCALATION: Needs Attention items with no assigned owner escalate to the service delivery lead.
```

Output: Delta-focused service health narrative for non-technical stakeholders.

Validate: Change items confirmed as genuine deltas vs the prior snapshot.

Next: Unowned attention items escalate to the service delivery lead.

LEVEL L1 HUMAN GATE No

SVC-06 Major Incident Communication

Use when: A major/critical incident requires timely, accurate stakeholder communication during the event.

Outcome: A communication draft that is honest about current knowledge, avoiding overpromising resolution time.

Inputs: Current incident status, [AUDIENCE], known facts vs unknowns, [DATE]/time of update

PROMPT

```
CONTEXT: Drafting major incident communication for <incident> affecting [CUSTOMER]/[AUDIENCE],
update as of [DATE]/time.
OBJECTIVE: Communicate current status honestly, distinguishing confirmed facts from ongoing
investigation, without overpromising resolution time.
INPUTS: Current incident status, confirmed impact, investigation progress, any resolution time
estimate basis.
CONSTRAINTS: Do not state a resolution ETA without a stated basis (e.g., "based on similar past
incidents" or "root cause identified, fix in progress") — an unfounded ETA damages credibility
more than "still investigating."
DECISION RULES: If root cause is unconfirmed, state "under investigation" explicitly rather than
implying a fix timeline.
OUTPUT CONTRACT: Sections — Current Status | Confirmed Impact | What We Know | What We Don't Know
Yet | Next Update Time | Point of Contact.
VALIDATION: Confirm no resolution ETA appears without an explicit stated basis.
ESCALATION: This is a live-incident communication artifact — route per incident communication
protocol immediately.
```

Output: Honest, fact-distinguishing major incident communication draft.

Validate: Confirmed no resolution ETA stated without an explicit basis.

Next: Issue at the next scheduled update time; repeat per protocol.

LEVEL L1 HUMAN GATE Yes

SVC-07 Capacity Trend Analysis (Service Ops)

Use when: Reviewing operational capacity trends (ticket volume, staffing) to plan ahead of demand shifts.

Outcome: A capacity trend analysis projecting when current staffing/capacity will be insufficient.

Inputs: Historical ticket/demand volume, current staffing/capacity, [THRESHOLD] service level target

PROMPT

```
CONTEXT: Analyzing operational capacity trends for <service desk/operations team> for [CUSTOMER].
OBJECTIVE: Project when current staffing/capacity will be insufficient to meet [THRESHOLD]
service levels, based on actual demand trend.
INPUTS: Historical ticket/demand volume, current staffing levels and average handling time,
target service level.
CONSTRAINTS: Project from the actual historical demand trend, not a flat assumption — state the
growth/seasonality basis used.
DECISION RULES: Flag AT RISK if projected demand will exceed current capacity's ability to meet
[THRESHOLD] within the planning horizon.
OUTPUT CONTRACT: Table [Period | Projected Demand | Current Capacity | Service Level Projection |
Flag] + recommended staffing action and lead time needed.
VALIDATION: Projected demand figures recalculated from the stated historical trend/growth basis.
ESCALATION: AT RISK findings within the hiring/training lead-time window route to staffing
decision-makers immediately.
```

Output: Trend-based operational capacity projection with a staffing action recommendation.

Validate: Projected demand figures recalculated from the stated trend basis.

Next: Immediate escalation to staffing decision-makers if within lead-time window.

LEVEL L1 HUMAN GATE No

SVC-08 Continual Service Improvement Plan

Use when: Defining a structured CSI plan from recurring service pain points, not ad hoc fixes.

Outcome: A CSI plan with each improvement tied to a specific recurring pain point and measurable target.

Inputs: Recurring incident/problem/SLA data, [SUCCESS_CRITERIA] for improvement, [CONSTRAINTS]

PROMPT

```
CONTEXT: Defining the continual service improvement plan for <service> for [CUSTOMER].
OBJECTIVE: Convert recurring incident/problem/SLA pain points into specific improvement
initiatives with measurable targets.
INPUTS: Recurring incident/problem data (from SVC-02), SLA performance trend (SVC-04),
improvement targets.
CONSTRAINTS: Every improvement initiative must trace to a specific, named recurring pain point
with supporting frequency data — not a general "improve reliability" statement.
DECISION RULES: Prioritize initiatives addressing the highest-frequency or highest-customer-
impact recurring problem first.
OUTPUT CONTRACT: Table [Pain Point | Frequency/Impact Evidence | Improvement Initiative | Target
Metric | Target Value | Owner | Review Date].
VALIDATION: Confirm each initiative cites the specific pain point and frequency evidence driving
it.
ESCALATION: Initiatives requiring investment beyond operational budget route to service delivery
leadership for approval.
```

Output: Evidence-traced CSI plan with measurable targets and owners.

Validate: Each initiative confirmed to cite specific pain-point frequency evidence.

Next: Review progress against targets at each stated review date.

LEVEL L2 HUMAN GATE Yes

Customer & Engagement

8 Prompts Across 1 Practitioner Role

Customer and engagement work begins **before a solution is proposed**. I designed these prompts to help customer-facing practitioners move from conversations and stated needs toward a **clear, evidence-grounded understanding of the problem, its business impact, solution feasibility, and value proposition**.

Role	Prompts	Core Focus
Customer / Engagement / Presales (PRE)	**8**	Discovery, problem definition, feasibility, value proposition, qualification & engagement decisions

From Customer Conversation to Solution Opportunity

The PRE prompts are designed to bring discipline to the early stages of an engagement. They help separate the **customer's actual problem from prematurely proposed solutions**, establish the business drivers behind the need, and identify what is still unknown before solutioning begins. Where business impact can be quantified, the prompts require the figures to be grounded in customer-provided evidence rather than assumptions.

This creates a more credible progression:

DISCOVER → DEFINE → ASSESS → POSITION → QUALIFY

The approach is particularly useful for shaping discovery outputs, validating whether a proposed approach is realistically deliverable within known constraints, and building a value proposition without overstating benefits. Feasibility is treated as an explicit step precisely to avoid **overselling a solution that cannot realistically be delivered**.

Evidence Before Promise

For customer-facing work, credibility matters as much as speed. I therefore designed these prompts to distinguish **what the customer has stated, what the available evidence demonstrates, and what still needs to be confirmed**. When information is insufficient, the appropriate response is to expose the gap and drive the next discovery question—not manufacture certainty.

The result is a compact set of **8 prompts that help transform customer conversations into better-qualified opportunities, more defensible solution discussions, and stronger foundations for delivery.**

Customer / Engagement / Presales (PRE)

Discovery, feasibility and proposal analysis grounded in evidence, avoiding overcommitment.

PRE-01 Discovery Session Synthesis

Use when: Converting raw customer discovery session notes into a structured problem/opportunity summary.

Outcome: A synthesized discovery summary distinguishing stated customer problems from solution-jumping.

Inputs: [SOURCE_DATA] discovery session notes, [CUSTOMER] context

PROMPT

```
CONTEXT: Synthesizing discovery session notes for [CUSTOMER] following a recent engagement
session.
OBJECTIVE: Extract the customer's actual stated problems and business drivers, separate from any
solution ideas raised prematurely.
INPUTS: Raw session notes/transcript, known customer context.
CONSTRAINTS: Distinguish PROBLEM (a stated pain/need) from SOLUTION IDEA (a proposed fix,
possibly premature) — do not let early solution talk obscure the underlying problem.
DECISION RULES: Capture a problem statement only where the customer's own words or clearly
attributed context support it — do not infer unstated pain points.
OUTPUT CONTRACT: Sections — Stated Problems (with customer quote/paraphrase source) | Business
Drivers | Solution Ideas Raised (parked, not yet validated) | Open Questions for Next Session.
VALIDATION: Confirm every stated problem traces to a specific point in the session notes.
ESCALATION: Problems implying budget/authority uncertainty route to PRE-02/PRE-08 for a
qualification check.
```

Output: Source-traced discovery summary separating problems from premature solutions.

Validate: Every stated problem traced to a specific point in the session notes.

Next: Feed into PRE-02 Customer Problem Definition.

LEVEL L1 HUMAN GATE No

PRE-02 Customer Problem Definition

Use when: Converting synthesized discovery input into a crisp, validated problem definition for solutioning.

Outcome: A validated problem definition with quantified business impact where evidence supports it.

Inputs: Discovery synthesis (PRE-01), any customer-provided data/metrics, [BUSINESS_OUTCOME]

PROMPT

```
CONTEXT: Defining the core customer problem for [CUSTOMER] to guide solution design toward
[BUSINESS_OUTCOME].
OBJECTIVE: Produce a crisp problem definition with quantified business impact where customer-
provided evidence supports it.
INPUTS: Discovery synthesis, any customer metrics/data shared, stated business outcome sought.
CONSTRAINTS: Quantify impact only using customer-provided or customer-confirmed figures — do not
estimate financial impact on the customer's behalf without their input.
DECISION RULES: If impact cannot be quantified from available data, state it qualitatively and
flag DATA GAP — do not fabricate a figure to make the business case look stronger.
OUTPUT CONTRACT: Sections — Problem Statement | Quantified Impact (or DATA GAP) | Root Cause
(customer's own view) | Why Now | Success Definition (customer's words where possible).
VALIDATION: Confirm every quantified figure traces to a customer-provided source, not an
assumption.
ESCALATION: Problem definitions lacking any quantified impact route back to PRE-01 for a follow-
up discovery question before proposal work proceeds.
```

Output: Evidence-grounded problem definition with quantified impact or an explicit gap.

Validate: Every quantified figure traced to a customer-provided source.

Next: Feed into PRE-03 Feasibility Assessment and PRE-07 Value Proposition.

LEVEL L1 HUMAN GATE No

PRE-03 Feasibility Assessment (Presales)

Use when: Before proposing a solution, assessing whether it is genuinely deliverable within likely customer constraints.

Outcome: An honest feasibility read that avoids overselling a solution beyond what's realistically deliverable.

Inputs: Problem definition (PRE-02), known [CONSTRAINTS] (timeline, budget signals, technical landscape)

PROMPT

```
CONTEXT: Assessing feasibility of a proposed solution approach for [CUSTOMER]'s problem: <state
problem> within known [CONSTRAINTS].
OBJECTIVE: Give an honest feasibility read to avoid committing to something undeliverable in the
proposal stage.
INPUTS: Problem definition, known technical landscape/constraints, indicative timeline/budget
signals if shared.
CONSTRAINTS: Do not rate FEASIBLE if a material technical or organizational blocker is known but
unaddressed — presales optimism bias is a named risk to guard against here.
DECISION RULES: CONDITIONAL if feasible only with a specific named dependency (customer data
readiness, a technical integration, budget tier); INFEASIBLE if a hard blocker exists with no
workaround.
OUTPUT CONTRACT: Sections — Feasibility Verdict | Known Blockers/Dependencies | Conditions for
Success | Confidence Level (with basis).
VALIDATION: Confirm any CONDITIONAL/INFEASIBLE verdict names the specific blocking factor.
ESCALATION: INFEASIBLE verdicts route to the deal team for a scope or expectation reset before
proposal submission.
```

Output: Honest, blocker-specific feasibility verdict for presales scoping.

Validate: Conditional/infeasible verdicts confirmed to name the specific blocking factor.

Next: Feasible/conditional items proceed to PRE-04/PRE-07.

LEVEL L1 HUMAN GATE Yes

PRE-04 RFP / RFI Response Analysis

Use when: An RFP/RFI has landed and needs structured analysis before committing to respond.

Outcome: A structured RFP analysis with a clear bid/no-bid input and identified response risk areas.

Inputs: RFP/RFI document, [CONSTRAINTS] (capability fit, timeline), known competitive context

PROMPT

```
CONTEXT: Analyzing an RFP/RFI from <customer/prospect> for a bid/no-bid decision.
OBJECTIVE: Assess requirement fit, identify response risk areas (compliance gaps, unrealistic
timeline, scope ambiguity) and support a bid/no-bid decision.
INPUTS: RFP/RFI document, current capability/reference fit, response timeline and resourcing
constraints.
CONSTRAINTS: Flag any mandatory/compliance requirement not currently met — do not gloss over a
hard requirement gap in the bid/no-bid framing.
DECISION RULES: Recommend NO-BID if a mandatory requirement cannot be met and no credible path to
meet it exists within the response timeline; otherwise BID with named risk areas to address in
the response.
OUTPUT CONTRACT: Sections — Requirement Fit Summary | Mandatory Requirement Gaps (if any) |
Competitive Position Assessment | Response Risk Areas | Bid/No-Bid Recommendation.
VALIDATION: Confirm every mandatory requirement in the RFP is checked against actual current
capability, not assumed met.
ESCALATION: Bid decisions with material resourcing or pricing risk route to deal leadership for
sign-off.
```

Output: Requirement-checked RFP analysis with a clear bid/no-bid recommendation.

Validate: Every mandatory requirement confirmed checked against actual current capability.

Next: Bid decisions route to deal leadership for sign-off.

LEVEL L2 HUMAN GATE Yes

PRE-05 Proposal Risk Review

Use when: Before submitting a proposal, reviewing it for commitments that create delivery or commercial risk.

Outcome: A proposal risk review flagging specific overcommitments before they become contractual obligations.

Inputs: Draft proposal content, known delivery capability/capacity, [CONSTRAINTS]

PROMPT

```
CONTEXT: Reviewing the draft proposal for [CUSTOMER] before submission.
OBJECTIVE: Identify specific commitments (dates, scope, SLAs, pricing assumptions) that carry
delivery or commercial risk if accepted as written.
INPUTS: Draft proposal text, known delivery team capacity/capability, standard commercial risk
thresholds.
CONSTRAINTS: Flag a commitment as RISKY only where it conflicts with known capacity/capability
data or lacks a stated assumption/dependency — do not flag standard, well-supported commitments.
DECISION RULES: Flag HIGH RISK if a date/SLA commitment has no stated dependency or assumption
protecting it (e.g., "assumes customer data delivered by X"); flag PRICING RISK if scope is
described ambiguously enough to invite scope-creep disputes.
OUTPUT CONTRACT: Table [Proposal Section | Commitment | Risk Type | Rationale | Suggested
Mitigation (e.g., add assumption/dependency clause)].
VALIDATION: Confirm every HIGH RISK flag names the specific missing protective clause or capacity
conflict.
ESCALATION: Pricing/commercial risk items route to commercial/deal leadership before submission.
```

Output: Specific-commitment proposal risk review with suggested protective clauses.

Validate: High-risk flags confirmed to name the specific missing clause or capacity conflict.

Next: Commercial risk items route to deal leadership before submission.

LEVEL L1 HUMAN GATE Yes

PRE-06 POC Success Criteria Design

Use when: Defining objective success criteria for a customer proof-of-concept before it starts.

Outcome: Pre-committed, measurable POC success criteria agreed before execution, preventing post-hoc goalpost-moving.

Inputs: Customer problem definition (PRE-02), what the POC will test, [SUCCESS_CRITERIA] discussion with customer

PROMPT

```
CONTEXT: Designing POC success criteria for [CUSTOMER]'s evaluation of <solution/technology>.
OBJECTIVE: Define specific, measurable success criteria agreed with the customer before the POC
starts, preventing later disagreement.
INPUTS: Problem definition, what the POC will and will not test (explicit scope), any success
expectations the customer has voiced.
CONSTRAINTS: Every criterion must be measurable and have a stated pass threshold — "customer is
happy with results" is not acceptable output.
DECISION RULES: A criterion belongs IN SCOPE for the POC only if it can be validly tested within
the POC's timeframe and environment; otherwise it becomes a Phase 2/production validation item,
stated explicitly.
OUTPUT CONTRACT: Sections — POC Scope (What's Tested / What's Not) | Success Criteria (each with
measurable threshold) | Sign-off Process | Post-POC Decision Path.
VALIDATION: Confirm every success criterion has a stated, specific pass threshold.
ESCALATION: Criteria requiring customer data/environment access route to a data-sharing agreement
before POC start.
```

Output: Pre-agreed, measurable POC success criteria with explicit scope boundaries.

Validate: Every success criterion confirmed to have a specific, stated pass threshold.

Next: Evaluate results via SA-11 POC Evaluation once complete.

LEVEL L1 HUMAN GATE Yes

PRE-07 Value Proposition / Business Case

Use when: Building the value proposition/business case to support a customer's internal investment decision.

Outcome: A business case quantifying value using customer-validated figures, avoiding inflated benefit claims.

Inputs: Problem definition (PRE-02), proposed solution, customer-provided cost/benefit data

PROMPT

```
CONTEXT: Building the value proposition/business case for [CUSTOMER] to support their internal
investment decision on <solution>.
OBJECTIVE: Quantify expected value using customer-validated data, avoiding inflated or generic
industry-benchmark claims presented as customer-specific.
INPUTS: Problem definition with quantified impact (PRE-02), proposed solution scope, any
customer-provided baseline costs/metrics.
CONSTRAINTS: Label any industry-benchmark-derived figure clearly as a benchmark, not a customer-
specific guarantee — conflating the two undermines credibility and sets unrealistic expectations.
DECISION RULES: Present value ranges (conservative/expected) rather than a single optimistic
figure, unless the customer's own data supports high-confidence precision.
OUTPUT CONTRACT: Sections — Current State Cost/Impact (customer data) | Proposed Solution Value
(Conservative/Expected range, with basis for each figure) | Investment Required | Payback Period
| Assumptions Log.
VALIDATION: Confirm every value figure's source (customer data vs industry benchmark) is
explicitly labeled.
ESCALATION: Business cases used in formal customer procurement processes route to deal leadership
for review before submission.
```

Output: Source-labeled business case with conservative/expected value ranges.

Validate: Every value figure's source explicitly labeled as customer data or benchmark.

Next: Formal procurement submissions route to deal leadership for review.

LEVEL L2 HUMAN GATE Yes

PRE-08 Customer Engagement Risk Assessment

Use when: Assessing the health and risk of a customer engagement/deal before a key milestone or decision.

Outcome: An engagement risk assessment naming specific risk signals, not a general sentiment read.

Inputs: Engagement history/interactions, stakeholder engagement level, [CONSTRAINTS] (timeline, budget signals)

PROMPT

```
CONTEXT: Assessing engagement risk for the [CUSTOMER] deal/relationship ahead of <key
milestone/decision>.
OBJECTIVE: Identify specific, observable risk signals (stakeholder disengagement, budget
authority uncertainty, competitive activity, timeline slippage) rather than a general confidence
read.
INPUTS: Engagement/interaction history, meeting attendance and seniority patterns, any stated
budget/authority/timeline signals.
CONSTRAINTS: Base each risk flag on an observable signal (e.g., "economic buyer has not attended
last 2 sessions") — not a vague "feels like it's cooling."
DECISION RULES: Flag HIGH RISK if the economic buyer/decision-maker has disengaged, or if a
previously stated timeline has slipped twice with no new date committed.
OUTPUT CONTRACT: Table [Risk Signal | Observed Evidence | Risk Level | Recommended Action] +
overall engagement health read.
VALIDATION: Confirm every flagged risk cites a specific observable event, not a general
impression.
ESCALATION: HIGH RISK signals on strategic deals route to deal/account leadership for a re-
engagement plan.
```

Output: Signal-based customer engagement risk assessment with recommended actions.

Validate: Every flagged risk confirmed to cite a specific observable event.

Next: High-risk deals route to account leadership for a re-engagement plan.

LEVEL L1 HUMAN GATE Yes

AI Delivery & AI Architecture

18 Prompts Across 1 Practitioner Role

AI initiatives require more than selecting a model or adding a prompt to an existing process. I designed this section to address the **AI-native delivery layer**—from identifying genuine AI opportunities through business-case development, architecture, evaluation, governance, security, cost, adoption, and value realization. The role brings together product, architecture, and delivery perspectives so that AI solutions are treated as **business capabilities that must be designed, governed, operated, and measured**.

Role	Prompts	Core Focus
AI Product Manager / AI Architect / AI Delivery Lead (AIX)	**18**	AI use-case discovery, opportunity assessment, architecture, evaluation, governance, security, cost, adoption & value

From AI Opportunity to Realized Value

The prompts establish a practical progression across the AI lifecycle:

DISCOVER → ASSESS → DESIGN → EVALUATE → GOVERN → OPERATE → ADOPT → MEASURE

The starting point is deliberately **problem-first**. AI use cases are expected to trace back to documented business or process pain, have a plausible data foundation, and demonstrate a meaningful value opportunity. This prevents technology enthusiasm from becoming the justification for an AI initiative.

From there, the library addresses the decisions required to move an opportunity toward production: **business case, AI architecture, model and capability choices, evaluation, governance, security, cost optimization, and operating considerations**. The architecture perspective is equally important—the right solution may involve a model, retrieval, workflow, agent, or a combination, depending on the work profile and risk.

Build for Adoption and Measurable Outcomes

An AI solution is not successful simply because it is technically deployed. The section therefore extends beyond implementation into **adoption and value realization**. Actual usage must be distinguished from realized business benefit, with measured outcomes used to determine whether the original business case is being achieved.

This makes the 18 prompts useful across the full journey—from **“Should we use AI here?”** to **“Is the AI capability delivering the value we expected?”**

The objective is to help AI leaders move from AI experimentation to governed, economically viable, measurable business capability.

AI Product Manager / AI Architect / AI Delivery Lead (AIX)

AI use-case discovery through architecture, evaluation, governance and operating model design — the AI-native delivery layer.

AIX-01 AI Use-Case Discovery

Use when: Scanning a business domain for genuine AI opportunity, avoiding solution-first thinking.

Outcome: A ranked list of candidate AI use cases grounded in actual process pain, not technology enthusiasm.

Inputs: [BUSINESS_OUTCOME], process pain points ([SOURCE_DATA]), [CONSTRAINTS] (data availability, risk tolerance)

PROMPT

```
CONTEXT: Discovering AI use-case candidates for <domain/function> in service of
[BUSINESS_OUTCOME].
OBJECTIVE: Identify candidate AI use cases grounded in documented process pain, ranked by value
and feasibility, not technology novelty.
INPUTS: Process pain points/inefficiencies (from BA-08/BA-09 style evidence), available data
sources, risk tolerance for the domain.
CONSTRAINTS: A candidate qualifies only if it addresses a documented pain point with a plausible
data source to ground it — reject "AI for AI's sake" candidates with no evidenced problem.
DECISION RULES: Rank by (value potential × data readiness ÷ implementation complexity); flag
candidates in regulated/high-stakes decision domains for extra governance scrutiny regardless of
score.
OUTPUT CONTRACT: Table [Candidate Use Case | Pain Point Addressed | Data Source Available? |
Value Potential | Complexity | Governance Sensitivity | Rank].
VALIDATION: Confirm each candidate traces to a specific documented pain point, not a generic
"could use AI here."
ESCALATION: High governance-sensitivity candidates (legal, medical, financial decisions affecting
individuals) route to AIX-12 AI Governance Framework before proceeding to business case.
```

Output: Ranked, pain-point-traced AI use-case candidate list.

Validate: Each candidate confirmed traced to a specific documented pain point.

Next: Top-ranked candidates feed AIX-02 Opportunity Assessment.

LEVEL L1 HUMAN GATE No

AIX-02 AI Opportunity Assessment / Business Case

Use when: A candidate AI use case needs a rigorous business case before architecture investment.

Outcome: A business case quantifying value, cost and risk of a specific AI use case, resisting hype-driven estimates.

Inputs: Candidate use case (AIX-01), [SUCCESS_CRITERIA], baseline process cost/performance data

PROMPT

```
CONTEXT: Building the business case for AI use case: <state use case> targeting
[SUCCESS_CRITERIA].
OBJECTIVE: Quantify expected value, implementation cost (including ongoing model/inference cost)
and risk, using baseline data.
INPUTS: Baseline process cost/performance (current state), use case scope, known constraints.
CONSTRAINTS: Include ongoing operating cost (inference, monitoring, human review capacity) in the
cost estimate — a business case showing only build cost is incomplete and must be flagged as
such.
DECISION RULES: Recommend PROCEED if expected value exceeds total cost of ownership within an
acceptable payback window and no ungoverned high-stakes risk exists; otherwise PILOT FIRST or DO
NOT PROCEED with rationale.
OUTPUT CONTRACT: Sections — Baseline State (quantified) | Expected Value (Conservative/Expected
range) | Total Cost of Ownership (build + ongoing) | Risk Summary | Payback Period |
Recommendation.
VALIDATION: Confirm the cost figure includes ongoing operating cost, not build cost alone.
ESCALATION: High-stakes or ungoverned-risk use cases route to AIX-12 Governance Framework before
funding approval.
```

Output: Full-lifecycle-cost AI business case with a clear proceed/pilot/decline recommendation.

Validate: Cost figure confirmed to include ongoing operating cost, not build cost alone.

Next: Approved cases proceed to AIX-03 AI Architecture Design.

LEVEL L2 HUMAN GATE Yes

AIX-03 AI Architecture Design

Use when: Designing the technical architecture for an approved AI use case, choosing the simplest effective pattern.

Outcome: An AI architecture design justified against the work profile (per R10: simplest effective architecture).

Inputs: Use case requirements, work profile (volume, ambiguity, determinism), [CONSTRAINTS]

PROMPT

```
CONTEXT: Designing AI architecture for <use case> on [PROJECT], within [CONSTRAINTS].
OBJECTIVE: Select the simplest architecture pattern (single-prompt/copilot, prompt chain,
workflow, agent, multi-agent) that meets the requirement — do not default to the most
sophisticated option.
INPUTS: Use case requirements, work characteristics (volume, ambiguity, determinism, need for
tool use), latency/cost constraints.
CONSTRAINTS: Justify any architecture more complex than a single prompt/copilot explicitly —
added complexity (chains, agents, multi-agent) must be earned by a specific requirement it
uniquely satisfies.
DECISION RULES: Use single-prompt/copilot for low-ambiguity, human-supervised tasks; workflow for
deterministic multi-step processes; agent only where dynamic tool selection/adaptive execution is
genuinely required; multi-agent only where distinct specialist reasoning domains are both
required and cannot be handled by a single well-scoped agent.
OUTPUT CONTRACT: Sections — Work Profile Summary | Architecture Pattern Selected with Rationale |
Model Routing Approach (see AIX-05) | Data/Context Sources | Human Checkpoint Placement |
Rejected Simpler Alternatives and Why They're Insufficient.
VALIDATION: Confirm the design explicitly shows why the simplest pattern (one level down in
complexity) was rejected, not just why the chosen one works.
ESCALATION: Agent/multi-agent designs touching consequential outputs (R9 categories) route to
AIX-13 Human-in-the-Loop Design before build.
```

Output: Complexity-justified AI architecture design with rejected-simpler-alternative rationale.

Validate: Design explicitly shows why the next-simpler pattern was insufficient.

Next: Consequential-output designs route to AIX-13 before build.

LEVEL L2 HUMAN GATE Yes

AIX-04 Model Selection Matrix

Use when: Choosing which model(s) to use for a use case or component, based on task fit rather than brand preference.

Outcome: A model selection matrix scoring candidates on task-relevant criteria: quality, latency, cost, context needs.

Inputs: Task requirements (reasoning depth, latency tolerance, volume), candidate models, [CONSTRAINTS] (budget)

PROMPT

```
CONTEXT: Selecting model(s) for <task/component> within the AI architecture for [PROJECT].
OBJECTIVE: Score candidate models against task-relevant criteria — reasoning/quality need,
latency tolerance, expected volume, cost per call — not general capability reputation alone.
INPUTS: Task requirements (ambiguity/reasoning depth, response time needs, expected call volume),
candidate model options, budget constraints.
CONSTRAINTS: Do not default to the highest-capability model for high-volume, low-ambiguity tasks
where a faster/cheaper model meets quality bar — cost/throughput matters for these profiles per
R10.
DECISION RULES: High-volume/low-ambiguity → optimize for cost/throughput (fast model);
complex/ambiguous reasoning → optimize for quality/judgment (reasoning model); mixed workloads →
recommend model routing (AIX-05).
OUTPUT CONTRACT: Table [Model Candidate | Quality Fit | Latency | Cost per Call | Volume Fit |
Recommendation] + final selection with rationale.
VALIDATION: Confirm the final selection's rationale explicitly references the task's
volume/ambiguity profile, not brand preference.
ESCALATION: Selections for consequential/high-risk output categories (R9) require the human-
review workflow regardless of model chosen.
```

Output: Task-profile-matched model selection with explicit rationale.

Validate: Final selection rationale confirmed to reference the actual volume/ambiguity profile.

Next: Mixed workloads route to AIX-05 Model Routing Strategy.

LEVEL L1 HUMAN GATE No

AIX-05 Model Routing Strategy

Use when: A system handles varied request types and needs a routing strategy to the right model/capability per request.

Outcome: A routing strategy with explicit, testable rules for directing requests to the appropriate model tier.

Inputs: Request type taxonomy, per-type volume and complexity profile, [CONSTRAINTS] (cost budget, latency SLA)

PROMPT

```
CONTEXT: Designing model routing strategy for <system> on [PROJECT] handling varied request
types.
OBJECTIVE: Define explicit routing rules directing each request type to the appropriately-sized
model/capability tier.
INPUTS: Request type taxonomy with volume and complexity per type, cost/latency budget.
CONSTRAINTS: Every routing rule must be testable/observable at request time (e.g., request
length, detected intent category, confidence score) — not a subjective judgment call embedded in
the router.
DECISION RULES: Route to fast/cheap tier by default; escalate to reasoning tier when a defined
trigger fires (e.g., low confidence score, detected ambiguity, explicit complexity signal, prior
tier's output failed validation).
OUTPUT CONTRACT: Table [Request Type/Trigger | Routed Tier | Rationale | Fallback if Tier Fails]
+ expected cost/latency profile under projected volume mix.
VALIDATION: Confirm every routing rule's trigger is observable/measurable at request time, not
subjective.
ESCALATION: Requests that fail routing/classification entirely should fall back to human
handling, not a default model guess — a named escalation path is mandatory.
```

Output: Testable model routing strategy with explicit triggers and fallback paths.

Validate: Every routing trigger confirmed observable/measurable at request time.

Next: Monitor routing accuracy; feed into AIX-09 AI Evaluation Framework.

LEVEL L3 HUMAN GATE No

AIX-06 RAG Architecture Design

Use when: Designing retrieval-augmented generation to ground AI outputs in an organization's own content.

Outcome: A RAG design addressing chunking, retrieval quality and access control, not just 'connect a vector DB.'

Inputs: Data readiness assessment (DA-08), grounding source(s), [SUCCESS_CRITERIA] for retrieval accuracy

PROMPT

```
CONTEXT: Designing RAG architecture for <use case> on [PROJECT], grounding source(s) confirmed
ready per DA-08.
OBJECTIVE: Define chunking strategy, retrieval approach and access-control preservation that
meets the retrieval accuracy target in [SUCCESS_CRITERIA].
INPUTS: Data readiness assessment, source content structure/format, target retrieval
accuracy/latency.
CONSTRAINTS: Access controls on the original content must be preserved through the retrieval
layer — a user must never retrieve content their underlying permissions would not allow them to
see directly.
DECISION RULES: Choose chunking strategy based on content structure (semantic/section-based
chunking for structured documents; smaller fixed-window chunking with overlap for unstructured
text) — justify against the actual source format, not a default chunk size.
OUTPUT CONTRACT: Sections — Chunking Strategy with Rationale | Retrieval Approach (embedding
model, top-k, re-ranking if used) | Access Control Enforcement Point | Evaluation Plan for
Retrieval Quality (link to AIX-11) | Freshness/Update Strategy.
VALIDATION: Confirm the access control enforcement point is explicitly named and testable, not
assumed inherited.
ESCALATION: Any design where access control cannot be technically preserved routes to Security
Architect before proceeding — this is a data leakage risk.
```

Output: Structure-justified RAG design with an explicit, testable access control enforcement point.

Validate: Access control enforcement point confirmed explicit and testable.

Next: Evaluate retrieval quality via AIX-11 Context Evaluation.

LEVEL L3 HUMAN GATE Yes

AIX-07 Agent Design Specification

Use when: An agent architecture has been justified (AIX-03) and needs a full instruction/behavior specification.

Outcome: A complete agent specification using the universal agent template, with explicit stop conditions.

Inputs: Agent's goal/scope, available tools, [CONSTRAINTS] (autonomy limits), escalation policy

PROMPT

```
CONTEXT: Specifying the agent for <task/goal> within [PROJECT]'s AI architecture (justified per
AIX-03).
OBJECTIVE: Produce a complete agent specification with explicit stop conditions and human
approval points — an agent without defined stop conditions is not production-ready.
INPUTS: Agent goal and scope boundary, available tools/actions, autonomy limits, escalation
policy for out-of-scope situations.
CONSTRAINTS: Every tool the agent can invoke must have a stated boundary on when it may be used —
do not grant blanket tool access without scoping.
DECISION RULES: Actions affecting customer, financial, security, legal, production or executive
outputs (per R9) require a human approval checkpoint before execution, not after-the-fact review.
OUTPUT CONTRACT: Follow the universal agent template — Goal | Context | Memory | Tools (with
scope per tool) | Decision Rules | Execution Steps | Validation | Stop Conditions (explicit list)
| Human Approval Points | Observability (what gets logged).
VALIDATION: Confirm every R9-category action has a pre-execution human approval checkpoint, not a
post-hoc review only.
ESCALATION: Agents with no reliable stop condition for a given failure mode must not be deployed
until one is defined — this is a hard gate, not a recommendation.
```

Output: Complete, stop-condition-explicit agent specification ready for build.

Validate: Every R9-category action confirmed to have a pre-execution approval checkpoint.

Next: Build and test against AIX-09 AI Evaluation Framework before production release.

LEVEL L4 HUMAN GATE Yes

AIX-08 Multi-Agent Architecture Design

Use when: A use case genuinely requires multiple specialist agents coordinating, justified beyond a single agent.

Outcome: A multi-agent design with explicit orchestration logic and failure-containment between agents.

Inputs: Specialist domains required, coordination pattern needs, [CONSTRAINTS]

PROMPT

```
CONTEXT: Designing multi-agent architecture for <use case> on [PROJECT], where single-agent
design (AIX-07) was assessed insufficient.
OBJECTIVE: Define the orchestration pattern (e.g., orchestrator-worker, sequential handoff,
debate/critique) and failure containment between agents.
INPUTS: Distinct specialist domains required and why they can't be handled by one well-scoped
agent, coordination/handoff requirements.
CONSTRAINTS: State explicitly why a single agent with broader instructions was insufficient —
multi-agent adds coordination failure modes and must be earned, not defaulted to for
"sophistication."
DECISION RULES: A failure in one agent must not silently propagate unchecked to downstream agents
— each handoff requires a validation checkpoint on the passed output.
OUTPUT CONTRACT: Sections — Why Single-Agent Was Insufficient | Orchestration Pattern | Agent
Roles & Boundaries | Handoff Validation Checkpoints | Failure Containment Strategy | Human
Approval Points (per R9).
VALIDATION: Confirm every inter-agent handoff has a stated validation checkpoint, not a direct
unchecked pass-through.
ESCALATION: Multi-agent systems touching consequential outputs require full AIX-13 Human-in-the-
Loop Design before production release.
```

Output: Orchestration-explicit multi-agent design with handoff validation checkpoints.

Validate: Every inter-agent handoff confirmed to have a stated validation checkpoint.

Next: Full human-in-the-loop design (AIX-13) required before production release.

LEVEL L4 HUMAN GATE Yes

AIX-09 AI Evaluation Framework

Use when: Defining how an AI system's output quality will be measured before and after production release.

Outcome: An evaluation framework with a labeled test set and explicit pass/fail thresholds per dimension.

Inputs: Use case success criteria, representative test cases, [THRESHOLD] for release quality

PROMPT

```
CONTEXT: Defining the evaluation framework for <AI system/component> on [PROJECT] prior to
release.
OBJECTIVE: Define measurable evaluation dimensions (accuracy, groundedness, safety, latency,
cost) with a labeled test set and explicit release thresholds.
INPUTS: Use case success criteria, representative real or synthetic test cases covering typical
and edge conditions, target quality threshold.
CONSTRAINTS: The test set must include adversarial/edge cases, not only typical happy-path
examples — evaluation on happy-path-only data overstates readiness.
DECISION RULES: Set a release threshold per dimension in [THRESHOLD]; the system is release-ready
only if every dimension clears its threshold, not an averaged composite score.
OUTPUT CONTRACT: Sections — Evaluation Dimensions (with measurement method) | Test Set
Composition (typical vs edge/adversarial split) | Per-Dimension Threshold | Scoring Method | Re-
evaluation Cadence (post-release drift check).
VALIDATION: Confirm the test set composition explicitly shows an edge/adversarial case
proportion, not zero.
ESCALATION: Dimensions that cannot be measured with available data are flagged DATA GAP — release
decision on that dimension requires explicit human risk acceptance.
```

Output: Edge-case-inclusive evaluation framework with per-dimension release thresholds.

Validate: Test set composition confirmed to include a non-zero edge/adversarial proportion.

Next: Run before release; re-run per the stated drift-check cadence.

LEVEL L2 HUMAN GATE Yes

AIX-10 Prompt Evaluation & Regression Testing

Use when: A prompt or prompt chain is being changed and needs regression testing before deployment.

Outcome: A regression test result confirming the prompt change doesn't degrade performance on prior-passing cases.

Inputs: Current prompt version, proposed change, existing test set (AIX-09), [THRESHOLD]

PROMPT

```
CONTEXT: Evaluating a proposed prompt change for <component> on [PROJECT] against the existing
regression test set.
OBJECTIVE: Confirm the change improves the targeted issue without degrading performance on
previously-passing cases.
INPUTS: Current and proposed prompt versions, existing labeled test set, the specific issue the
change targets.
CONSTRAINTS: Run the full existing test set against both versions, not just the cases motivating
the change — regressions on unrelated cases are the primary risk of prompt edits.
DECISION RULES: Approve the change only if it improves the targeted case(s) AND does not drop any
previously-passing case below its threshold; otherwise flag REGRESSION and hold.
OUTPUT CONTRACT: Table [Test Case | Prior Version Result | New Version Result | Delta | Flag] +
net verdict (Approve/Regression Found/Hold for Investigation).
VALIDATION: Confirm the full existing test set (not a subset) was run against both prompt
versions.
ESCALATION: Regressions on consequential-output test cases (R9 categories) block deployment
regardless of net improvement elsewhere.
```

Output: Full-regression-tested prompt change verdict with per-case deltas.

Validate: Confirmed the full existing test set, not a subset, was run against both versions.

Next: Approved changes deploy; regressions route back for prompt revision.

LEVEL L2 HUMAN GATE Conditional

AIX-11 Context Evaluation / Grounding Assessment

Use when: Assessing whether an AI system's retrieved/provided context is actually sufficient and accurate for the task.

Outcome: A grounding assessment identifying whether failures stem from retrieval quality or model reasoning.

Inputs: Sample of AI outputs with their retrieved/provided context, ground truth where available

PROMPT

```
CONTEXT: Assessing context/grounding quality for <AI system, e.g., RAG-based> on [PROJECT].
OBJECTIVE: Determine whether observed output quality issues stem from insufficient/incorrect
retrieved context (a retrieval problem) or from the model reasoning poorly given good context (a
model/prompt problem) — these require different fixes.
INPUTS: Sample of outputs with the exact context that was retrieved/provided for each, ground
truth or expert judgment on correctness.
CONSTRAINTS: Classify each failure case specifically — RETRIEVAL FAILURE (correct answer was not
present in the provided context) vs REASONING FAILURE (correct answer was present in context but
the model didn't use it correctly) — do not treat all failures as one category.
DECISION RULES: If retrieval failure rate exceeds [THRESHOLD], prioritize AIX-06 RAG design fixes
(chunking, retrieval method); if reasoning failure rate is dominant, prioritize AIX-10 prompt
evaluation/model selection.
OUTPUT CONTRACT: Table [Case | Failure Type | Context Provided (summary) | Expected Answer |
Model Output | Root Cause Category] + failure type distribution summary.
VALIDATION: Confirm each classification checked the actual provided context against the expected
answer, not assumed.
ESCALATION: Dominant retrieval failure rate routes to AIX-06 for a RAG architecture revisit;
dominant reasoning failure routes to AIX-04/AIX-10.
```

Output: Retrieval-vs-reasoning failure classification with a distribution summary.

Validate: Each classification confirmed checked against the actual provided context.

Next: Route fixes to AIX-06 (retrieval) or AIX-04/AIX-10 (reasoning) per the dominant failure type.

LEVEL L2 HUMAN GATE No

AIX-12 AI Governance Framework

Use when: Establishing organizational policy for how AI use cases are approved, monitored and controlled.

Outcome: A governance framework with tiered approval requirements matched to use-case risk, not one-size-fits-all.

Inputs: Portfolio of AI use cases/candidates, [RISK_TOLERANCE], regulatory context if applicable

PROMPT

```
CONTEXT: Establishing AI governance framework for [CUSTOMER]/[PROGRAM] covering the AI use-case
portfolio.
OBJECTIVE: Define risk-tiered approval, monitoring and human-oversight requirements matched to
each use case's actual stakes.
INPUTS: Use-case portfolio with their domains (e.g., internal productivity vs customer-facing
decisions), applicable regulatory context.
CONSTRAINTS: Tiering must be based on actual decision stakes (impact on individuals,
financial/legal/safety consequence) — not on technical sophistication of the AI approach.
DECISION RULES: HIGH-STAKES tier (affects individual rights, financial outcomes, safety, legal
standing) requires mandatory human-in-the-loop (AIX-13) and pre-release evaluation sign-off (AIX-
09); LOW-STAKES internal productivity tools require lighter-touch monitoring only.
OUTPUT CONTRACT: Sections — Risk Tiers (definition + examples) | Approval Requirements per Tier |
Monitoring Requirements per Tier | Human Oversight Requirements per Tier | Incident/Escalation
Process for AI-caused harm.
VALIDATION: Confirm tiering criteria are based on decision stakes, not AI technique
sophistication.
ESCALATION: Use cases affecting legal, safety, or protected-class outcomes route to
legal/compliance for review regardless of internal risk tier assignment.
```

Output: Stakes-tiered AI governance framework with per-tier approval and oversight requirements.

Validate: Tiering criteria confirmed based on decision stakes, not technique sophistication.

Next: Apply per use case via AIX-01/AIX-02 intake screening.

LEVEL L2 HUMAN GATE Yes

AIX-13 Human-in-the-Loop Design

Use when: Designing exactly where and how human review/approval integrates into an AI workflow.

Outcome: A human-in-the-loop design specifying checkpoint placement, reviewer authority and override mechanics.

Inputs: AI workflow/agent design (AIX-03/07/08), governance tier (AIX-12), [CONSTRAINTS] (reviewer capacity)

PROMPT

```
CONTEXT: Designing human-in-the-loop checkpoints for <AI workflow/agent> on [PROJECT], governance tier per AIX-12.
OBJECTIVE: Specify exactly where human review sits in the workflow, what authority the reviewer has, and how overrides are captured for future improvement.
INPUTS: AI workflow/agent design, governance tier requirements, available reviewer capacity.
CONSTRAINTS: A checkpoint that reviewers cannot meaningfully act on within their available time is not a real control — size checkpoint frequency/volume to actual reviewer capacity, or flag the capacity gap explicitly.
DECISION RULES: Place checkpoints BEFORE execution for R9-category consequential actions (irreversible or high-impact); AFTER execution (sampled review) is acceptable only for low-stakes, reversible actions.
OUTPUT CONTRACT: Sections — Checkpoint Placement (pre/post, with rationale) | Reviewer Authority (approve/reject/edit/escalate) | Override Capture Mechanism (feeds back into AIX-10 regression set) | Reviewer Capacity Check | Fallback if Reviewer Unavailable.
VALIDATION: Confirm every pre-execution checkpoint is confirmed for a genuinely irreversible/high-impact action, and reviewer capacity is checked against actual expected volume.
ESCALATION: Capacity gaps (more review volume than reviewers can handle) route to delivery leadership for staffing or scope reduction — do not silently reduce review rigor to fit capacity.
```

Output: Checkpoint-explicit human-in-the-loop design with a reviewer capacity check.

Validate: Reviewer capacity confirmed checked against actual expected review volume.

Next: Capacity gaps escalate to delivery leadership before go-live.

LEVEL L2 HUMAN GATE Yes

AIX-14 AI Security Assessment

Use when: Assessing an AI system for AI-specific security risks (prompt injection, data exfiltration via tool use, model misuse).

Outcome: An AI-specific security assessment covering injection, exfiltration and misuse vectors, not generic app-sec only.

Inputs: AI system architecture (tools/data access it has), [SYSTEMS] it can act on, data sensitivity

PROMPT

```
CONTEXT: Assessing AI-specific security risk for <AI system/agent> on [PROJECT] with access to [SYSTEMS].
OBJECTIVE: Identify prompt injection, data exfiltration and tool-misuse vectors specific to AI systems, beyond standard application security.
INPUTS: System architecture including what tools/data the AI can access, trust level of input sources (e.g., untrusted user content, third-party documents fed into context), data sensitivity.
CONSTRAINTS: Evaluate every input source the AI processes for injection risk, including indirect sources (e.g., a document or webpage the AI reads as part of a task) — not just direct user chat input.
DECISION RULES: Flag HIGH RISK if the AI can take a consequential action (per R9) based on instructions found in untrusted content (indirect prompt injection surface), or if it has tool access that could exfiltrate sensitive data without a human checkpoint.
OUTPUT CONTRACT: Table [Vector | Description | Untrusted Input Source | Consequential Action Reachable? | Risk Level | Mitigation].
VALIDATION: Confirm every input source the AI processes was evaluated, including indirect/tool-retrieved content, not chat input alone.
ESCALATION: HIGH RISK findings route to Security Architect and block production release until mitigated or explicitly risk-accepted by security leadership.
```

Output: Injection- and exfiltration-focused AI security assessment with named mitigations.

Validate: Confirmed every input source, including indirect/tool-retrieved content, was evaluated.

Next: High-risk findings block release pending SEC sign-off.

LEVEL L2 HUMAN GATE Yes

AIX-15 AI Cost Optimization Analysis

Use when: AI inference/operating costs need to be reduced without degrading output quality below threshold.

Outcome: A cost optimization plan quantifying savings per lever with quality impact stated, not assumed neutral.

Inputs: Current AI system cost breakdown (per call, by component), usage volume, [THRESHOLD] quality floor

PROMPT

```
CONTEXT: Optimizing AI operating cost for <system> on [PROJECT], maintaining quality above
[THRESHOLD].
OBJECTIVE: Identify specific cost levers (model routing/downsizing, prompt/context length
reduction, caching, batching) with quantified savings and stated quality impact.
INPUTS: Current cost breakdown by component/call type, usage volume, current measured quality
baseline.
CONSTRAINTS: Every cost lever must state its expected impact on quality (even if "none expected,
to be verified") — do not present savings without addressing the quality trade-off risk.
DECISION RULES: Recommend caching for repeated/similar queries with stable answers; recommend
model downsizing only where AIX-04-style task profile supports a smaller model meeting the
quality floor; recommend context/prompt trimming only where evaluation (AIX-09) confirms no
groundedness loss.
OUTPUT CONTRACT: Table [Lever | Affected Component | Estimated Savings | Expected Quality Impact
| Verification Method Before Rollout] + total estimated savings.
VALIDATION: Confirm every lever includes a stated verification method to confirm no quality
regression before full rollout.
ESCALATION: Levers affecting consequential-output components (R9) require AIX-09/AIX-10 re-
evaluation before deployment, not just a spot-check.
```

Output: Quality-verified AI cost optimization plan with quantified savings per lever.

Validate: Every lever confirmed to include a stated pre-rollout verification method.

Next: Verify via AIX-09/AIX-10 before full rollout of any lever.

LEVEL L2 HUMAN GATE Conditional

AIX-16 AI ROI Analysis

Use when: Post-deployment, measuring actual realized ROI of an AI initiative against the original business case.

Outcome: A realized-ROI analysis against the original business case (AIX-02), using measured data, not re-estimated projections.

Inputs: Original business case (AIX-02), actual measured usage/outcome data post-deployment

PROMPT

```
CONTEXT: Measuring realized ROI for <AI initiative> on [PROJECT] post-deployment, against the
original business case.
OBJECTIVE: Compare actual measured value and cost against the original projections,
distinguishing realized from still-projected benefit.
INPUTS: Original business case figures, actual measured adoption/usage, actual measured operating
cost, actual measured outcome change (e.g., time saved, error reduction) where instrumented.
CONSTRAINTS: Only claim REALIZED value where a measured outcome exists — usage/adoption alone is
not value realization unless tied to a measured downstream outcome.
DECISION RULES: Mark benefit REALIZED if measured outcome data confirms it; PARTIALLY REALIZED if
adoption is happening but outcome measurement is incomplete; NOT REALIZED if adoption is low or
outcome data shows no change.
OUTPUT CONTRACT: Table [Original Projected Benefit | Measured Actual | Status | Evidence Source]
+ actual operating cost vs projected + net ROI verdict.
VALIDATION: Every REALIZED status backed by a cited measurement source, not adoption metrics
alone.
ESCALATION: NOT REALIZED verdicts route to sponsor and AIX-17 Adoption Plan review — a technical
success with no realized ROI is an adoption problem, not a build problem.
```

Output: Measured, evidence-graded AI ROI verdict against the original business case.

Validate: Every realized status backed by a cited measurement source, not adoption alone.

Next: Not-realized verdicts route to AIX-17 Adoption & Change Plan for review.

LEVEL L2 HUMAN GATE No

AIX-17 AI Adoption & Change Plan

Use when: An AI capability is built but adoption is lagging, or a new capability needs a deliberate adoption plan.

Outcome: An adoption plan addressing the specific, evidenced barrier to adoption, not a generic training push.

Inputs: Current adoption data/barriers, [AUDIENCE] (target users), [SUCCESS_CRITERIA] for adoption

PROMPT

```
CONTEXT: Building the adoption/change plan for <AI capability> targeting [AUDIENCE], current
adoption status: <describe>.
OBJECTIVE: Diagnose the specific barrier to adoption (trust, workflow fit, skill gap, unclear
value) and design an intervention matched to it.
INPUTS: Adoption/usage data, direct user feedback if available, workflow context the capability
sits within.
CONSTRAINTS: Diagnose the barrier from actual evidence (usage drop-off point, feedback themes)
before designing the intervention — do not default to "more training" without evidence that skill
gap is the actual barrier.
DECISION RULES: Trust barrier → transparency/explainability intervention (show reasoning, allow
easy override); workflow-fit barrier → redesign integration points, not user retraining; skill
gap → targeted enablement; unclear value → re-communicate with concrete before/after examples.
OUTPUT CONTRACT: Sections — Adoption Barrier Diagnosis (with evidence) | Matched Intervention |
Success Metric for Adoption | Review Cadence.
VALIDATION: Confirm the intervention type matches the diagnosed barrier per the decision rule,
not a generic default.
ESCALATION: Trust barriers tied to a specific past AI failure/incident route to AIX-09 to confirm
evaluation coverage before re-promoting adoption.
```

Output: Evidence-diagnosed adoption barrier with a matched intervention plan.

Validate: Intervention type confirmed matched to the diagnosed barrier, not a generic default.

Next: Trust-barrier cases route to AIX-09 to confirm evaluation coverage.

LEVEL L2 HUMAN GATE No

AIX-18 AI Operating Model / Workforce Architecture Design

Use when: Designing how human roles, AI systems and oversight structures fit together at an organizational level.

Outcome: An operating model defining human-AI role boundaries and oversight structure, not just a tooling rollout plan.

Inputs: Target processes/functions, current role structure, [BUSINESS_OUTCOME], governance framework (AIX-12)

PROMPT

```
CONTEXT: Designing the AI operating model / workforce architecture for <function/domain> under
[PROGRAM], targeting [BUSINESS_OUTCOME].
OBJECTIVE: Define which tasks shift to AI, which remain human, which become human-AI
collaborative, and the resulting oversight/accountability structure.
INPUTS: Current process/role breakdown, target state ambition, governance framework tiering (AIX-
12).
CONSTRAINTS: Every task reallocation to AI must state the human accountability structure that
remains — AI does not remove accountability, it relocates the point of human judgment; this must
be explicit, not implied.
DECISION RULES: Tasks in HIGH-STAKES governance tiers remain human-decision with AI-assist only
(per AIX-12/AIX-13); LOW-STAKES, high-volume tasks may shift to AI-primary with human-exception-
handling; MEDIUM tasks become human-AI collaborative with defined handoff points.
OUTPUT CONTRACT: Sections — Current State Role/Task Map | Target State Role/Task Map (Human/AI-
Assist/AI-Primary per task, with governance tier) | Accountability Structure Post-Change |
Skill/Role Transition Plan | Oversight Cadence.
VALIDATION: Confirm every AI-Primary task has an explicit human accountability owner named, not
left implicit.
ESCALATION: Role/workforce structure changes with people impact route to HR/people leadership and
require executive sponsor sign-off before communication.
```

Output: Accountability-explicit AI operating model with a human/AI task allocation map.

Validate: Every AI-primary task confirmed to have an explicit named human accountability owner.

Next: People-impact changes route to HR/executive sponsor before communication.

LEVEL L3 **HUMAN GATE Yes**

Part 3 — Prompt Index

All 206 prompts in document order. Use Ctrl+F / Cmd+F to search by ID, keyword or role.

ID	Prompt Name	Role	Lvl	Gate
TDM-01	Delivery Health Snapshot	Technical Delivery Manager	L1	No
TDM-02	Multi-Workstream Status Roll-up	Technical Delivery Manager	L1	No
TDM-03	RAID Log Triage & Refresh	Technical Delivery Manager	L1	No
TDM-04	Risk Exposure Scoring	Technical Delivery Manager	L1	Conditional
TDM-05	Dependency Cascade Analysis	Technical Delivery Manager	L2	Conditional
TDM-06	Critical Path Impact Assessment	Technical Delivery Manager	L1	No
TDM-07	Milestone Slippage Diagnosis	Technical Delivery Manager	L2	No
TDM-08	Resource & Capacity Risk Assessment	Technical Delivery Manager	L1	Conditional
TDM-09	Technical Debt Delivery Impact	Technical Delivery Manager	L2	No
TDM-10	Escalation Brief Generator	Technical Delivery Manager	L1	Yes
TDM-11	Recovery Plan Builder	Technical Delivery Manager	L2	Yes
TDM-12	Steering Committee Narrative	Technical Delivery Manager	L1	Yes
TDM-13	Executive Decision Brief	Technical Delivery Manager	L1	Yes
TDM-14	Vendor / Partner Performance Review	Technical Delivery Manager	L2	Yes
TDM-15	Weekly Delivery Intelligence Digest	Technical Delivery Manager	L1	No
PM-01	Scope Baseline & Boundary Definition	Project Manager	L1	Yes
PM-02	Schedule Health Check	Project Manager	L1	No
PM-03	Cost / Budget Variance Analysis	Project Manager	L1	Conditional
PM-04	RAID Log Build	Project Manager	L1	No
PM-05	Risk Response Planning	Project Manager	L2	Conditional
PM-06	Change Request Impact Analysis	Project Manager	L1	Yes
PM-07	Resource Plan & Allocation Check	Project Manager	L1	No
PM-08	Stakeholder Analysis & Engagement Plan	Project Manager	L1	Conditional
PM-09	Governance Cadence Design	Project Manager	L1	Yes
PM-10	Status Report Generator	Project Manager	L1	No
PM-11	Recovery Plan	Project Manager	L2	Yes
PM-12	Benefits & Lessons Learned Review	Project Manager	L2	Conditional
TPM-01	Cross-Project Dependency Map	Program Manager / TPM	L2	No
TPM-02	Integrated Roadmap Builder	Program Manager / TPM	L2	Yes
TPM-03	Program Risk Aggregation	Program Manager / TPM	L2	No
TPM-04	Program Critical Path Analysis	Program Manager / TPM	L2	No
TPM-05	Program Capacity Analysis	Program Manager / TPM	L2	No
TPM-06	Executive Program Report	Program Manager / TPM	L1	Yes

ID	Prompt Name	Role	Lvl	Gate
TPM-07	Program Governance Design	Program Manager / TPM	L1	Yes
TPM-08	Trade-off Analysis	Program Manager / TPM	L1	Yes
TPM-09	Benefits Tracking (Program Level)	Program Manager / TPM	L2	No
TPM-10	Scenario Planning	Program Manager / TPM	L2	Yes
TPM-11	Program Escalation Brief	Program Manager / TPM	L1	Yes
PMO-01	Portfolio Health Assessment	PMO / Transformation Lead	L2	No
PMO-02	Governance Framework Design	PMO / Transformation Lead	L1	Yes
PMO-03	KPI / Metrics Framework	PMO / Transformation Lead	L1	No
PMO-04	PMO Maturity Assessment	PMO / Transformation Lead	L2	No
PMO-05	Transformation Roadmap	PMO / Transformation Lead	L2	Yes
PMO-06	Benefits Realization Framework	PMO / Transformation Lead	L2	Yes
PMO-07	Portfolio Risk Aggregation	PMO / Transformation Lead	L2	No
PMO-08	Executive Portfolio Report	PMO / Transformation Lead	L1	Yes
PMO-09	Governance Automation Candidate Assessment	PMO / Transformation Lead	L1	No
PMO-10	AI Adoption Readiness Assessment	PMO / Transformation Lead	L2	No
SM-01	Iteration Health Diagnostic	Scrum Master / Agile Practitioner	L1	No
SM-02	Impediment Analysis & Removal Plan	Scrum Master / Agile Practitioner	L1	No
SM-03	Retrospective Synthesis & Action Plan	Scrum Master / Agile Practitioner	L1	No
SM-04	Backlog Health Assessment	Scrum Master / Agile Practitioner	L1	No
SM-05	Flow Metrics Analysis	Scrum Master / Agile Practitioner	L1	No
SM-06	Team Health Assessment	Scrum Master / Agile Practitioner	L1	Yes
SM-07	Anti-Pattern Detection	Scrum Master / Agile Practitioner	L2	Yes
SM-08	Predictability & Forecast Analysis	Scrum Master / Agile Practitioner	L1	Conditional
SM-09	Coaching Plan Builder	Scrum Master / Agile Practitioner	L1	Conditional
SM-10	Release Readiness Assessment	Scrum Master / Agile Practitioner	L1	Yes
SM-11	Cross-Team Dependency Management	Scrum Master / Agile Practitioner	L1	No
SM-12	SAFe PI / ART Sync Adapter	Scrum Master / Agile Practitioner	L1	No
SM-13	Kanban Flow Adapter	Scrum Master / Agile Practitioner	L1	No
SM-14	XP Engineering Practice Adapter	Scrum Master / Agile Practitioner	L1	No
SM-15	Lean / LeSS Large-Scale Adapter	Scrum Master / Agile Practitioner	L1	No
SM-16	Nexus / DASM Tailoring Adapter	Scrum Master / Agile Practitioner	L1	Yes
PO-01	Customer Problem Discovery Synthesis	Product Manager / Product Owner	L1	No
PO-02	Product Requirements Definition	Product Manager / Product Owner	L1	Yes
PO-03	Prioritization Decision (RICE)	Product Manager / Product Owner	L1	No
PO-04	Roadmap Trade-off Analysis	Product Manager / Product Owner	L1	Yes

ID	Prompt Name	Role	Lvl	Gate
PO-05	Experiment Design & Hypothesis	Product Manager / Product Owner	L2	Conditional
PO-06	Outcome Metrics Framework	Product Manager / Product Owner	L1	No
PO-07	Value vs Effort Assessment	Product Manager / Product Owner	L1	No
PO-08	WSJF Scoring Model	Product Manager / Product Owner	L1	No
PO-09	MVP Scope Definition	Product Manager / Product Owner	L1	Yes
PO-10	Product Decision Brief	Product Manager / Product Owner	L1	Yes
BA-01	Requirements Discovery Plan	Business Analyst	L1	No
BA-02	Requirements Decomposition	Business Analyst	L1	No
BA-03	Ambiguity & Conflict Resolution	Business Analyst	L1	Conditional
BA-04	User Story Writing	Business Analyst	L1	No
BA-05	Acceptance Criteria Definition	Business Analyst	L1	No
BA-06	Business Rules Extraction	Business Analyst	L1	No
BA-07	Non-Functional Requirements Definition	Business Analyst	L1	No
BA-08	Process Mapping (As-Is / To-Be)	Business Analyst	L2	No
BA-09	Gap Analysis	Business Analyst	L1	No
BA-10	Stakeholder Analysis (BA View)	Business Analyst	L1	No
BA-11	Requirements Traceability Matrix	Business Analyst	L2	No
BA-12	Impact Analysis	Business Analyst	L2	No
TL-01	Technical Design Review	Technical Lead / Engineering Lead	L1	Yes
TL-02	Code Review Assistant	Technical Lead / Engineering Lead	L1	Yes
TL-03	Technical Debt Assessment	Technical Lead / Engineering Lead	L1	No
TL-04	Engineering Risk Assessment	Technical Lead / Engineering Lead	L1	No
TL-05	API Design Review	Technical Lead / Engineering Lead	L1	Conditional
TL-06	Integration Design Analysis	Technical Lead / Engineering Lead	L1	No
TL-07	Scalability Assessment	Technical Lead / Engineering Lead	L2	No
TL-08	Performance Bottleneck Analysis	Technical Lead / Engineering Lead	L1	No
TL-09	Reliability Assessment	Technical Lead / Engineering Lead	L2	No
TL-10	CI/CD Pipeline Assessment	Technical Lead / Engineering Lead	L1	No
TL-11	Engineering Estimation	Technical Lead / Engineering Lead	L1	No
TL-12	Build-vs-Buy Analysis	Technical Lead / Engineering Lead	L1	Yes
SA-01	Architecture Options Analysis	Solution Architect	L2	Yes
SA-02	Architecture Trade-off Matrix	Solution Architect	L1	Yes
SA-03	NFR Definition & Validation	Solution Architect	L2	Yes
SA-04	Integration Architecture Design	Solution Architect	L2	No
SA-05	API Strategy Design	Solution Architect	L2	Yes
SA-06	Scalability Architecture Review	Solution Architect	L2	No

ID	Prompt Name	Role	Lvl	Gate
SA-07	Reliability Architecture Review	Solution Architect	L2	No
SA-08	Security Architecture Review (Solution-Level)	Solution Architect	L2	Yes
SA-09	Technology Selection Matrix	Solution Architect	L1	Yes
SA-10	Feasibility Assessment	Solution Architect	L1	Yes
SA-11	POC Evaluation	Solution Architect	L1	Yes
SA-12	Architecture Decision Record (ADR) Generator	Solution Architect	L1	No
CA-01	Cloud Architecture Design	Cloud Architect	L2	No
CA-02	Migration Strategy & Wave Plan	Cloud Architect	L2	Yes
CA-03	Modernization Assessment	Cloud Architect	L2	Yes
CA-04	Landing Zone Design	Cloud Architect	L2	Yes
CA-05	Well-Architected Review	Cloud Architect	L2	No
CA-06	Cloud Cost Optimization Analysis	Cloud Architect	L1	Conditional
CA-07	Resilience / HA Design	Cloud Architect	L2	No
CA-08	DR Strategy Design	Cloud Architect	L2	Yes
CA-09	IAM & Security Posture Review	Cloud Architect	L1	Yes
CA-10	Observability Strategy Design	Cloud Architect	L2	No
IA-01	Capacity Planning Analysis	Infrastructure Architect	L1	No
IA-02	Network Architecture Review	Infrastructure Architect	L2	No
IA-03	Compute Sizing Analysis	Infrastructure Architect	L1	No
IA-04	Storage Architecture Review	Infrastructure Architect	L1	No
IA-05	HA/DR Design Review	Infrastructure Architect	L2	No
IA-06	Backup Strategy Review	Infrastructure Architect	L1	Yes
IA-07	Migration Readiness Assessment	Infrastructure Architect	L1	No
IA-08	Hybrid Infrastructure Design	Infrastructure Architect	L2	Yes
EA-01	Capability Mapping	Enterprise Architect	L2	No
EA-02	Application Portfolio Rationalization	Enterprise Architect	L2	Yes
EA-03	Technology Standards Assessment	Enterprise Architect	L1	Conditional
EA-04	Target Architecture Definition	Enterprise Architect	L2	Yes
EA-05	Architecture Principles Development	Enterprise Architect	L1	Yes
EA-06	Transformation Roadmap	Enterprise Architect	L2	Yes
EA-07	Modernization Business Case	Enterprise Architect	L2	Yes
EA-08	Architecture Governance Review	Enterprise Architect	L2	Yes
SEC-01	Threat Modeling	Security Architect	L2	Yes
SEC-02	Security Architecture Review	Security Architect	L2	Yes
SEC-03	IAM Design Review	Security Architect	L1	Yes

ID	Prompt Name	Role	Lvl	Gate
SEC-04	Zero Trust Readiness Assessment	Security Architect	L2	No
SEC-05	Security Controls Gap Analysis	Security Architect	L2	Conditional
SEC-06	Security Risk Assessment	Security Architect	L1	Yes
SEC-07	Vulnerability Prioritization	Security Architect	L1	Conditional
SEC-08	Compliance Mapping	Security Architect	L2	Yes
DA-01	Data Architecture Design	Data Architect	L2	No
DA-02	Data Modeling Review	Data Architect	L1	No
DA-03	Data Governance Framework	Data Architect	L2	Yes
DA-04	Data Quality Assessment	Data Architect	L1	Conditional
DA-05	Data Lineage Mapping	Data Architect	L2	No
DA-06	Data Migration Strategy	Data Architect	L2	Yes
DA-07	Analytics Architecture Design	Data Architect	L2	No
DA-08	AI / RAG Data Readiness Assessment	Data Architect	L2	Yes
OPS-01	CI/CD Pipeline Design Review	DevOps / SRE / Platform Engineer	L2	Conditional
OPS-02	Deployment Risk Assessment	DevOps / SRE / Platform Engineer	L1	Yes
OPS-03	Incident Response Analysis	DevOps / SRE / Platform Engineer	L1	No
OPS-04	SLO/SLA Definition	DevOps / SRE / Platform Engineer	L1	Yes
OPS-05	Error Budget Analysis	DevOps / SRE / Platform Engineer	L1	Conditional
OPS-06	Observability Strategy (Platform-Level)	DevOps / SRE / Platform Engineer	L2	No
OPS-07	Reliability Engineering Review	DevOps / SRE / Platform Engineer	L2	Conditional
OPS-08	Capacity & Scaling Analysis	DevOps / SRE / Platform Engineer	L1	No
OPS-09	Root Cause Analysis (RCA)	DevOps / SRE / Platform Engineer	L2	No
OPS-10	Platform Automation Candidate Assessment	DevOps / SRE / Platform Engineer	L1	No
QA-01	Test Strategy Design	QA / Test Lead	L2	No
QA-02	Test Plan Generator	QA / Test Lead	L1	No
QA-03	Requirements-to-Test Traceability	QA / Test Lead	L1	No
QA-04	Risk-Based Test Prioritization	QA / Test Lead	L1	Conditional
QA-05	Regression Suite Optimization	QA / Test Lead	L2	Yes
QA-06	Defect Trend Analysis	QA / Test Lead	L1	No
QA-07	Test Automation Candidate Assessment	QA / Test Lead	L1	No
QA-08	Performance Test Strategy	QA / Test Lead	L2	No
QA-09	Release Quality Gate Assessment	QA / Test Lead	L1	Yes
QA-10	Test Coverage Gap Analysis	QA / Test Lead	L1	No
SVC-01	Incident Analysis	Service Delivery / IT Operations	L1	No
SVC-02	Problem Management (RCA)	Service Delivery / IT Operations	L2	No
SVC-03	Change Risk Assessment	Service Delivery / IT Operations	L1	Yes

ID	Prompt Name	Role	Lvl	Gate
SVC-04	SLA Performance Review	Service Delivery / IT Operations	L1	Conditional
SVC-05	Service Health Dashboard Narrative	Service Delivery / IT Operations	L1	No
SVC-06	Major Incident Communication	Service Delivery / IT Operations	L1	Yes
SVC-07	Capacity Trend Analysis (Service Ops)	Service Delivery / IT Operations	L1	No
SVC-08	Continual Service Improvement Plan	Service Delivery / IT Operations	L2	Yes
PRE-01	Discovery Session Synthesis	Customer / Engagement / Presales	L1	No
PRE-02	Customer Problem Definition	Customer / Engagement / Presales	L1	No
PRE-03	Feasibility Assessment (Presales)	Customer / Engagement / Presales	L1	Yes
PRE-04	RFP / RFI Response Analysis	Customer / Engagement / Presales	L2	Yes
PRE-05	Proposal Risk Review	Customer / Engagement / Presales	L1	Yes
PRE-06	POC Success Criteria Design	Customer / Engagement / Presales	L1	Yes
PRE-07	Value Proposition / Business Case	Customer / Engagement / Presales	L2	Yes
PRE-08	Customer Engagement Risk Assessment	Customer / Engagement / Presales	L1	Yes
AIX-01	AI Use-Case Discovery	AI Product Manager / AI Architect / AI Delivery Lead	L1	No
AIX-02	AI Opportunity Assessment / Business Case	AI Product Manager / AI Architect / AI Delivery Lead	L2	Yes
AIX-03	AI Architecture Design	AI Product Manager / AI Architect / AI Delivery Lead	L2	Yes
AIX-04	Model Selection Matrix	AI Product Manager / AI Architect / AI Delivery Lead	L1	No
AIX-05	Model Routing Strategy	AI Product Manager / AI Architect / AI Delivery Lead	L3	No
AIX-06	RAG Architecture Design	AI Product Manager / AI Architect / AI Delivery Lead	L3	Yes
AIX-07	Agent Design Specification	AI Product Manager / AI Architect / AI Delivery Lead	L4	Yes
AIX-08	Multi-Agent Architecture Design	AI Product Manager / AI Architect / AI Delivery Lead	L4	Yes
AIX-09	AI Evaluation Framework	AI Product Manager / AI Architect / AI Delivery Lead	L2	Yes
AIX-10	Prompt Evaluation & Regression Testing	AI Product Manager / AI Architect / AI Delivery Lead	L2	Conditional
AIX-11	Context Evaluation / Grounding Assessment	AI Product Manager / AI Architect / AI Delivery Lead	L2	No
AIX-12	AI Governance Framework	AI Product Manager / AI Architect / AI Delivery Lead	L2	Yes
AIX-13	Human-in-the-Loop Design	AI Product Manager / AI Architect / AI Delivery Lead	L2	Yes
AIX-14	AI Security Assessment	AI Product Manager / AI Architect / AI Delivery Lead	L2	Yes

ID	Prompt Name	Role	Lvl	Gate
AIX-15	AI Cost Optimization Analysis	AI Product Manager / AI Architect / AI Delivery Lead	L2	Conditional
AIX-16	AI ROI Analysis	AI Product Manager / AI Architect / AI Delivery Lead	L2	No
AIX-17	AI Adoption & Change Plan	AI Product Manager / AI Architect / AI Delivery Lead	L2	No
AIX-18	AI Operating Model / Workforce Architecture Design	AI Product Manager / AI Architect / AI Delivery Lead	L3	Yes