AI-Augmented Large Capital Project Management

Systems Thinking, Probabilistic Estimating, and Project Controls as Claude Code Skills

Prof. Dr. Johannes Meier

12 June 2026 — Revised edition

Executive Summary

Large capital projects systematically underperform. Across a database of 16,000 projects, Oxford University researchers found that only 8.5 percent met their original cost and schedule targets, and a mere 0.5 percent delivered all promised benefits on time and on budget.[1] McKinsey's analysis of more than 300 projects exceeding one billion dollars in value found average cost overruns of 80 percent and schedule delays of 50 percent.[2] These are not outlier statistics; they describe the norm.

This paper argues that the underperformance has two distinct root causes, each addressable with structured AI support:

Before the Final Investment Decision (FID), projects fail by definition. Analysis of fifteen landmark failure cases, spanning nuclear power stations, rail programmes, carbon capture schemes, hospital IT, and mining projects, shows that projects are scoped as asset construction programmes when their success depends on systems that extend far beyond the built object: regulatory architecture, value-chain dependencies, integration economics, social licence, and governance coherence. When the definition omits these couplings, the project builds part of its failure in from the start.

After FID, projects fail by information. McKinsey's analysis of 48 deeply troubled megaprojects found that 73 percent of cost and schedule overruns were caused by failures of monitoring, escalation, and corrective action during execution, not by flawed original designs.[3] Project controls data is generated too slowly, integrated too rarely, and interpreted too inconsistently for management to act before damage accumulates.

This paper presents an integrated, deployable response: a suite of Claude Code skills and project scaffolding that spans the capital project lifecycle:

- `systems-thinking` — a pre-FID audit skill that stress-tests a project definition against fifteen research-derived heuristics and produces a board-ready report with a 1–5 maturity score, a ranked gap list in a standardised omission vocabulary, and a recommended tool stack. In structured evaluation, outputs produced with the skill passed 100 percent of assertions, against 41 percent without it.
- `budget-estimator` — a probabilistic CAPEX skill that converts parametric work-package definitions into P50/P90 estimates via 10,000-iteration Monte Carlo simulation, with AACE-class-calibrated accuracy bounds and distribution-shaped contingency.
- `evm` — an execution-phase Earned Value Management skill (ANSI/EIA-748 compliant) that converts raw cost and schedule data into ten financial metrics per WBS package, three EAC forecast methods, five publication-ready charts, and automatic escalation flags, in minutes rather than days.
- `gate-readiness` — a stage-gate governance skill that checks the project's evidence base against an explicit, phase-calibrated checklist (estimate class, audit maturity, regulatory map, kill

criteria) and issues a READY / CONDITIONALLY READY / NOT READY assessment with a closure plan, so that approving bodies decide on the state of the evidence rather than on a narrative about it.
- `lessons-learned` — a close-out skill that compares final actuals against the original P50/P90 estimate, attributes the variance to scope, productivity, risk events, and escalation, and proposes versioned, provenance-tracked updates to the heuristics library, so that each completed project becomes calibration data for the next one.
- **Project scaffolding** — a `CLAUDE.md` AI constitution encoding judgment and escalation rules, versioned domain heuristics with mandatory provenance, JSON schema validation, and three specialist agents (schedule analysis, contract review, risk assessment).

The system is not a replacement for experienced project professionals. It is a force multiplier: it ensures that the same fifteen systemic tests are applied to every project definition before FID, that earned value calculations are performed consistently every reporting period, that escalation thresholds are enforced automatically, and that domain heuristics from decades of benchmarking are applied without relying on institutional memory.

The system has been designed, implemented, and tested on a hypothetical 500 km hydrogen transmission pipeline (EUR 1.75 billion CAPEX). It is available as open-source code at github.com/jmeier1963/large_capital_project_management.

---

1. The Capital Project Performance Crisis

1.1 The Scale of the Problem

Flyvbjerg's landmark study, the largest empirical analysis of project performance ever conducted, established what he calls the "iron law of megaprojects": over budget, over time, over and over again.[4] The headline numbers bear repeating:

- **91.5%** of projects exceed their original budget or schedule, or both
- **8.5%** deliver within original parameters
- **0.5%** deliver cost, schedule, *and* promised benefits

McKinsey adds sectoral granularity. Rail projects overrun by an average of 45 percent. Bridges and tunnels by 35 percent. Mining and metals projects, among the most capital-intensive category, see 83 percent of major projects exceed planned CAPEX by more than 40 percent, with average delays of 20 to 30 months.[5]

The energy transition amplifies the stakes. The Hydrogen Council and McKinsey reported in 2025 that the global hydrogen sector has now committed over USD 110 billion across more than 500 projects past Final Investment Decision.[6] These are projects where schedule delays translate directly into stranded capital, missed carbon commitments, and competitive disadvantage in a market where first-mover timing is commercially decisive.

These figures, while sobering, describe symptoms rather than causes. The causes divide into two phases of the project lifecycle, and each demands a different analytical response.

1.2 Failure Mechanism One: Systemic Definition Gaps Before FID

The deeper pattern, visible in the best-documented failure cases, is that projects are scoped as asset construction programmes when their success depends on systems that extend far beyond the built object. A nuclear power station requires integrated quality assurance, subcontractor oversight, and regulatory documentation as an indivisible whole, not reactor installation alone. A carbon capture project needs financeable full-chain economics spanning carbon pricing, transport tariffs, and storage contracts, beyond the capture unit itself. A hospital IT programme depends as much on clinical workflow integration, trust autonomy, and local change capability as on the software.

When the scope definition omits these dependencies, projects build part of their failure in from the start.

Analysis of fifteen landmark projects (selected because authoritative public post-mortems exist in sufficient detail to support root-cause attribution) reveals five recurring systemic blindspots:[7]

Blindspot 1: Legal-regulatory non-integration. The Longannet CCS project in the United Kingdom was cancelled when it became clear that funding caps, the carbon price floor, and the contract architecture formed an incompatible economic system that could not be made financeable. The regulatory environment was not a boundary condition to be managed; it was a structural component of the project's value thesis, and it was modelled too late. Pascua-Lama in Chile and Argentina followed the same logic: environmental regulation and indigenous rights were treated as downstream permitting tasks until court decisions made the project unbuildable.[8]

Blindspot 2: Build-before-design-freeze. Olkiluoto 3, Flamanville 3, Vogtle, and the Sydney Opera House are four variants of one pattern: political or commercial pressure for a visible start meets insufficiently mature design, incompletely validated manufacturing paths, or immature supply chains. The result goes far beyond rework: it is a qualitatively different project with newly created dependencies. At Olkiluoto 3, the design-fabrication-quality assurance loop was not treated as an integrated system. STUK investigations found that subcontractor control and regulatory documentation were separated from technical design in a way that made early cost and schedule estimates unrecoverable.[9]

Blindspot 3: Late integration economics. Crossrail and the NHS National Programme for IT both featured many subsystems that appeared "green" in isolation while the overall system remained non-operational. For Crossrail, civil works were mastered years ahead of signalling, station systems, safety assurance, and the thousands of physical-digital asset interactions that defined operational readiness. The NAO investigation found that the integration and testing phase had been systematically planned too small and started too late.[10] NPfIT collapsed for the equivalent reason in software: centralised architecture, heterogeneous legacy trust systems, and clinical workflow diversity were not integrated ends of the same delivery equation.

Blindspot 4: Physical-social coupling. The Hallandsås Tunnel in Sweden took 23 years and cost roughly ten times its original budget. The root cause was not geological surprise alone. Geology, groundwater chemistry, environmental law, and local legitimacy were treated as separate streams rather than a coupled physical-legal-social system. Once toxic injection agents entered the groundwater, the project became politically redefinable in ways that no schedule or cost model had anticipated. Pascua-Lama replicated this pattern: the project's physical footprint and its legal footprint diverged irreconcilably.[11]

Blindspot 5: Governance fragmentation. The Scottish Parliament Building, the Big Dig in Boston, and Crossrail each suffered from accountability for scope, cost, quality, and interfaces

being distributed across organisations, contracts, and political levels in ways that made systemic risk invisible at any single node. The Holyrood Audit Committee found that brief, design, and procurement stabilised too late precisely because no single governance structure had authority across the full system boundary.[12]

These five blindspots are not coincident failures of project management rigour. They are structurally predictable consequences of defining complex sociotechnical systems as if they were asset construction projects. The implication is straightforward but uncomfortable: a project definition that describes the physical asset without modelling the full system (feedstock, grids, permits, tax and levy treatment, operating model, supply-chain readiness, social licence, and exit logic) is not a definition. It is an engineering brief for one component of a system whose other components have not been described.

The research is explicit on this point: **"The skill must never treat project definition as asset definition alone."**[13] This design principle drives the `systems-thinking` skill described in Section 3.

1.3 Failure Mechanism Two: The Execution Information Gap

The intuition that projects fail in execution because of poor engineering or inadequate planning is only partially correct. McKinsey's analysis of 48 deeply troubled megaprojects found that **73 percent of cost and schedule overruns were caused by poor execution**, meaning failures of monitoring, escalation, and corrective action rather than flawed original designs.[3]

The mechanism is predictable. Project data (cost actuals, schedule progress, change orders, risk events) is generated continuously in the field. But in most organisations, this data travels slowly: from site time-sheets and contractor invoices into cost control systems, then into spreadsheets maintained by cost engineers, then into narrative reports assembled for management review. By the time a negative trend is visible in a board report, the underlying pattern may have been developing for three or four months.

The consequence is that corrective actions are taken late, when the cost of recovery is high. A schedule slip that costs EUR 2 million to correct in month three may cost EUR 20 million by month nine, not because the underlying problem grew tenfold but because the window for low-cost intervention closed.

Consistent, rigorous project controls are the established antidote: earned value management (EVM), integrated schedule analysis, and systematic risk quantification. The Independent Project Analysis Group (IPA), whose proprietary database covers more than 20,000 capital projects, measures project performance against a Project Control Index (PCI). Projects with strong project controls consistently outperform their peers on cost, schedule, and operability outcomes.[14]

The barrier to universal adoption of rigorous project controls is not lack of awareness. It is capacity and consistency. A cost engineer maintaining full earned value analysis across eight WBS packages, three active contracts, and five reporting periods simultaneously, while also managing contractor queries, validating invoices, and preparing the monthly report, will under pressure abbreviate the analysis. Thresholds will not be checked. EAC forecasts will not be run across all three methods. The escalation that should have gone to the Project Director this week will go next week instead.

This is the specific gap that AI-augmented project controls addresses.

1.4 What Has Changed: The AI Inflection Point

For most of the past decade, AI in project management meant dashboards that required extensive data pipeline engineering, natural language processing tools that extracted text with limited accuracy, and predictive models that required months of proprietary training data before producing actionable output. The technology was real but the deployment cost was prohibitive for all but the largest programmes.

The shift since 2023 is qualitative, not incremental. Large language models (LLMs), and in particular AI agents that reason over structured data, apply domain-specific rules, call tools, and generate structured outputs, have changed what is achievable without bespoke development.

Three capabilities now combine in a way that was not previously available:

Structured reasoning over domain rules. An AI agent can be given a plain- English rules file (e.g., "if CPI drops below 0.85, generate a Project Director escalation memo; if SPI has been below 0.85 for two consecutive periods, issue a mandatory written notice") and will apply those rules with perfect consistency on every run. The same mechanism applies to definitional audit rules: a fifteen-rule systems-thinking framework, encoded once, is applied in the same sequence with the same vocabulary on every project definition, regardless of sector. The rules are not remembered from a training corpus; they are encoded explicitly and executed deterministically.

Tool use and computation. Modern AI agents can invoke Python scripts, read CSV files, write structured output, and generate charts as a native part of their workflow. The EVM module in this system runs a Python engine (ANSI/EIA-748 compliant, 713 lines) that computes ten financial metrics per WBS package, runs three EAC forecast methods, produces five publication-ready charts, and writes a complete markdown report, all as a single AI-orchestrated pipeline. The budget estimator runs a 10,000-iteration Monte Carlo simulation the same way.

Domain heuristics without domain re-training. Parametric cost benchmarks, productivity norms, and contingency ranges can be loaded into the AI's context at runtime as versioned YAML files, without fine-tuning or model retraining. When the AI applies a benchmark, it cites the source and version. When project data falls outside the benchmark's valid range, it flags this explicitly. The heuristics library is updated as the organisation accumulates actuals, which turns project experience into institutional knowledge that the AI accesses on the next run.

This is not a claim that AI understands project management in the way an experienced cost engineer or project director does. It is a more precise and more practically important claim: AI can perform the *routine, structured, rule-governed portion* of project definition review and project controls work with perfect consistency. That frees skilled professionals to focus on the *judgment-intensive* portions (contractor negotiation, root cause analysis, recovery planning, stakeholder strategy) where human expertise is irreplaceable.

Avoiding the enterprise AI trap. Deloitte's 2025–2026 State of AI survey found that while nearly 90 percent of companies have deployed AI in at least one business function, 94 percent report not seeing significant value from their investments.[15] The explanation is structural: most AI deployments target individual productivity (drafting emails, summarising documents) rather than the *process* level where value is concentrated.

In capital project management, value is concentrated in process consistency, not individual productivity. The question is not whether a cost engineer can produce a report faster with AI assistance. The question is whether every project receives the same complete systemic audit, the same prob-

abilistic estimate discipline, and the same threshold-checked EVM report, every time, regardless of the experience level of its team and regardless of whether it is at the FID gate or in month thirty-two of execution.

The approach described in this paper embeds AI into the process as a non-optional execution step, with encoded domain rules, parameterised thresholds, and mandatory output formats that cannot be abbreviated under pressure.

---

2. An Integrated Skill Suite Across the Project Lifecycle

2.1 Lifecycle Coverage

The five skills and the project scaffolding map onto the capital project lifecycle as follows:

Lifecycle phase	Failure mechanism addressed	Skill / component	Primary output
Concept / Pre-FEED / FEED	Systemic definition gaps	`systems-thinking`	15-rule audit, maturity score
Pre-FID estimating	False certainty in point budgets	`budget-estimator`	P50/P90 CAPEX, tornado chart
Every decision gate	Gates passed on incomplete evidence	`gate-readiness`	Gate evidence pack, readiness verdict
Execution	Slow, inconsistent controls	`evm` + agents	Monthly EV report, escalations
Close-out	Lessons paid for but not captured	`lessons-learned`	Calibrated heuristics, lessons register
All phases	Discretionary judgment	Scaffolding	Encoded rules every run

The skills are deliberately independent (each can be adopted alone) but they compound. A project that passes a systems-thinking audit at maturity 3+ enters execution with an interface register, a regulatory map, and explicit kill criteria that the execution-phase agents can then monitor. A budget estimate built on the same heuristics library that the EVM module later benchmarks against closes the loop between estimate and actuals. The `lessons-learned` skill makes that loop operational by feeding close-out actuals back into the heuristics library with full provenance, so the next project estimates from calibrated rather than inherited benchmarks.

2.2 Design Principles

The system is built on four principles that distinguish it from generic AI implementations:

Data first, conversation second. Project intelligence is derived from structured data (CSV files, YAML configuration, JSON schemas), not from narratives or slide decks. This makes outputs reproducible, auditable, and comparable across periods and across projects.

Provenance on every heuristic. Cost benchmarks, productivity norms, and contingency ranges are stored as versioned YAML files with mandatory `source:` and `valid_range:` fields. When the AI applies a benchmark, it cites the source. When the project falls outside the valid range, it flags this explicitly.

Rules encoded, not remembered. Escalation thresholds, required approvals, audit criteria, and judgment rules are encoded in configuration files: the `CLAUDE.md` project constitution for execution

rules and the sys-rules.md reference for definitional audit criteria. They are applied automatically on every run, not recalled from a briefing document.

Roles, not generics. Every communication or action recommendation produced by the AI addresses a specific named role from the project's organisation chart. It does not recommend that "the team" should act. It specifies that the Commercial Manager should review the change order, that the Project Director should receive the escalation memo, and that the cost register should be updated by the Cost Engineer within five business days.

2.3 System Architecture

The execution-phase system consists of five integrated layers. (The systems-thinking skill, which operates on project documents rather than the structured data store, has its own two-file architecture described in Section 3.3.)

Layer 1 — Project Data Store

A structured directory of CSV, YAML, JSON, and Markdown files representing the project's living state. The recommended directory layout mirrors standard project controls practice:

```
my-project/
+-- CLAUDE.md              # project AI constitution
+-- project-brief.md       # scope, FID CAPEX, milestones
+-- 01-contracts/
|   +-- contracts-register.csv
|   +-- change-orders/
+-- 02-schedule/
|   +-- milestones.csv     # planned/forecast dates, critical path
+-- 03-cost/
|   +-- evm-timephased.csv # one row per WBS per period
|   +-- evm-output/        # written by EVM skill each period
+-- 04-resources/
|   +-- resource-matrix.csv
+-- 05-risk/
    +-- risk-register.csv
```

All files are validated against JSON schemas on intake. The directory is version-controlled, meaning every change is logged and reversible.

The CLAUDE.md AI Constitution. The central configuration file that defines how the AI must behave on this project. A production CLAUDE.md for a capital project encodes mandatory judgment rules in plain English:

- *"If CPI drops below 0.85, immediately generate a Project Director escalation memo referencing the WBS package, the current CPI value, and the three-period CPI trend."*
- *"If SPI has been below 0.85 for two consecutive reporting periods, generate a mandatory written escalation notice. Do not wait for a third period."*
- *"Any change order exceeding 1% of the contract value triggers an independent cost review. Notify the Commercial Manager and the PMO lead."*
- *"At stage-gate FEED-to-EPC, the estimate must be AACE Class 3 or better. Flag any estimates submitted at Class 4 or 5 as non-compliant."*

It also defines stage-gate-specific standards, so the same system applies different rigor depending on the project phase:

Phase	Estimate Class	Contingency Range	Schedule Basis
Concept	Class 5	35–40%	±50%
Pre-FEED	Class 4	25–30%	±30%
FEED	Class 3	15–20%	±15%
Execution	Class 2	8–12%	±10%
Closeout	Class 1	3–5%	Actuals

These rules are applied on every AI run without exception. They cannot be abbreviated because the cost engineer is under pressure.

Layer 2 — Domain Heuristics Library

A curated library of parametric benchmarks and productivity norms, stored as versioned YAML files. Two files are included for hydrogen pipeline projects:

`heuristics/pipeline-cost.yaml` — installed cost per kilometre by pipe diameter and terrain type (abbreviated):

```yaml
# Source: IPA Benchmarking Database + DVGW project actuals (Western Europe)
# Valid for: onshore H2 pipelines, Germany / Western Europe, AACE Class 3-5
# Accuracy: ±20-30% (Class 3 equivalent)
# Version: 1.0 | Updated: 2026-01

base_cost_eur_per_km:
  DN300: 1_450_000
  DN400: 1_920_000  # H2-PIPE-DE-001 reference
  DN500: 2_650_000
  DN600: 3_400_000

terrain_factors:
  flat_agricultural: 1.00
  rolling_rural:     1.15
  urban_corridor:    1.60
  river_crossing_hdd: 2.50  # per crossing

h2_service_premium: 0.15
# +15% for H2-grade materials (HIC-tested, dry-gas seals)

contingency_by_class:
  class_5: [0.30, 0.50]      # conceptual estimate
  class_3: [0.15, 0.25]      # study estimate
  class_2: [0.10, 0.15]      # budget-quality estimate
  class_1: [0.05, 0.10]      # definitive estimate
```

When the AI applies this benchmark to a cost estimate, it cites the source and version, and flags any project falling outside the valid range (e.g., a DN900 pipeline or a project in a different region). `heuristics/productivity-norms.yaml` covers labour productivity for pipeline welding (joints per shift by diameter and welder grade), civil excavation (m^3/shift by terrain), mechanical erection, and E&I installation, with the same provenance structure.

Layer 3 — Specialist Agents

Domain-specific agents are defined as markdown files in `scaffolding/agents/`. Each agent specifies its trigger conditions, analysis steps, output format requirements, and escalation rules. They are installed by copying to `.claude/agents/` in the project directory, where Claude Code discovers and invokes them automatically on matching prompts.

Agent	Trigger	Output
EVM Analyst	Monthly cost/schedule cycle	EV report, charts, EAC, RAG, escalations
Schedule Analyzer	Milestone slip; low SPI	Schedule health, float, recovery options
Contract Reviewer	Change order or new contract	Commercial summary, flagged clauses
Risk Assessor	Monthly cycle; new risk	Top-10 EMV digest, escalations

The `schedule-analyzer` agent executes a defined sequence: read `milestones.csv` and flag any milestone with a forecast slip exceeding 14 days; calculate total float on the critical path from the latest schedule export; compare the current SPI trend against the project's historical SPI curve; generate a recovery options matrix with cost and schedule impact for each option. Its escalation rule is encoded in its definition: any slip greater than 60 days on a milestone marked `critical:\true` triggers a mandatory Project Director memo, regardless of whether the SPI threshold has been breached.

The `contract-reviewer` agent processes change orders against a structured extraction template: contract type, base value, CO value as a percentage of contract, entitlement basis (scope change vs. changed conditions), dispute resolution mechanism, and liquidated-damages exposure. It cross-references the CO value against the `CLAUDE.md` threshold and generates the independent cost review notification automatically when the threshold is exceeded.

Layer 4 — Claude Code Skills

Claude Code's skill system packages purpose-built Python tools and structured analytical behaviour for automatic invocation. Skills are installed by placing a directory in `~{}/.claude/skills/`, containing a `SKILL.md` behavioural instruction file and, where applicable, the Python implementation. Claude Code scans this directory at startup and invokes skills when user prompts match the trigger phrases defined in `SKILL.md`.

Skill	Source	Capability
evm	Purpose-built	EVM engine: 10 metrics, 5 charts, 3 EAC forecast methods
budget-estimator	Purpose-built	Parametric CAPEX with P50/P90 Monte Carlo simulation
systems-thinking	Purpose-built	15-rule pre-FID systemic audit with maturity scoring
gate-readiness	Purpose-built	Stage-gate evidence pack assembly and readiness verdict
lessons-learned	Purpose-built	Close-out review, variance attribution, heuristics calibration
pdf	Marketplace	Extract structured text from contracts and specifications

Skill	Source	Capability
pptx	Marketplace	Assemble progress presentation from EVM report and charts
csv-data-summarizer	Marketplace	Statistical summary of cost registers and change order logs
meeting-insights-analyzer	Marketplace	Extract action items and decisions from meeting notes

Skills activate automatically on matching trigger phrases, as described in Sections 3 through 5 below.

Layer 5 — JSON Schemas and Validation

Three JSON Schema files enforce data quality at intake:

- `schemas/project-brief.schema.json` — validates the YAML frontmatter of `project-brief.md`: required fields (`project_id`, `fid_date`, `approved_capex_eur`, `contingency_eur`), data types, and allowable values for `status` and `phase`.
- `schemas/contract.schema.json` — validates contract register entries: contract type (lump sum / EPCM / reimbursable / time-and-materials), required date fields, and mandatory completion of `ld_rate_per_day` for lump-sum contracts.
- `schemas/risk-register.schema.json` — validates risk register entries: probability (0–1 float), impact (EUR value), required mitigation owner, and mandatory EMV recalculation trigger when either probability or impact changes by more than 10 percent.

Schema validation runs automatically when Claude Code opens a project directory. Any invalid data file generates an immediate quality flag before analysis begins.

2.4 The Integration and Feedback Skills: Gate Readiness and Lessons Learned

The three analytical skills produce evidence; two further skills govern how that evidence is used. Both are behavioural skills (markdown instruction files without computational engines) because their value lies in enforcing a process, not in performing a calculation.

`gate-readiness` — **every gate decided on the state of the evidence.** Stage gates fail in two ways: projects pass on incomplete evidence (the gate becomes theatre), or gate preparation consumes weeks of manual document assembly. The gate-readiness skill audits the project data store against an explicit, phase-calibrated evidence matrix. At FEED-to-FID, for example, that means a systems-thinking audit at maturity 3 or better with all omission flags dispositioned, a P50/P90 estimate at AACE Class 3 or better and less than 90 days old, a regulatory map with named owners, an interface register, schema- valid risk register, documented kill criteria, and an outside-view reference class comparison. Each item is rated PRESENT, STALE, SUBSTANDARD, or ABSENT, and the skill issues a READY / CONDITIONALLY READY / NOT READY assessment with a closure plan for every deficiency. Two cross-checks are automatic: a budget request below the estimate's P50 is flagged (the project is asking for less than its own central estimate), and an omission flag carried open across two consecutive gates is reported as a governance finding in its own right. The skill integrates the other skills' outputs; it never re-performs their analysis, and the go/no-go decision remains with the humans at the gate.

`lessons-learned` — **each project becomes calibration data for the next.** Most organisations pay twice for the same lesson: once on the project that taught it, and again on the next project that never heard it. The lessons-learned skill performs the close-out feedback loop in five steps. It measures estimate accuracy (where the final actual landed in the original P50/P90 distribution, and whether the claimed AACE accuracy band held). It attributes the variance to scope change, productivity, materialised risks, escalation, and an honestly stated residual, because scope growth does not discredit a unit-rate benchmark, while productivity variance does. It then proposes calibrated updates to the `heuristics/*.yaml` library as explicit YAML diffs with version bumps and provenance updates, for human review rather than automatic write; a single project is one data point, and the skill proposes changes only when deviations exceed the benchmark's stated accuracy band or confirm a pattern from a prior project. Transferable lessons are recorded in a standardised register mapped to the fifteen SYS rules. Finally, the skill closes the loop on the audit framework itself: it reports which pre-FID omission flags actually materialised into cost or delay, and whether anything material happened that no SYS rule covers. That is the evidence base from which the fifteen-rule framework earns its sixteenth rule, or does not.

---

3. The Systems-Thinking Skill: Auditing the Project Definition

3.1 The Problem with Ad Hoc Review

Experienced project directors, supervisory board members, and independent technical advisers routinely identify systemic risks in project definitions. The challenge is that unstructured review is inconsistent, non-comparable, and dependent on the reviewer's particular experience base. A reviewer whose background is energy markets may rigorously interrogate pricing assumptions while missing interface management gaps. One with infrastructure experience may flag integration risks while overlooking social licence dynamics.

The result is that the quality of a pre-FID systemic review correlates strongly with the identity of the reviewer, not the characteristics of the project. This makes governance a function of personnel rather than process, an arrangement that is difficult to audit, impossible to mandate, and unlikely to scale across a portfolio.

3.2 What the Skill Adds

A Claude Code skill packages structured analytical behaviour that is applied consistently regardless of context. The `systems-thinking` skill does not add domain knowledge that an experienced reviewer lacks: both the skill-guided and unskilled configurations of Claude correctly identify that a lithium hydroxide project's business case depends on lithium prices, or that a Finnish first-of-a-kind refinery requires an environmental permit under the YVA procedure. The skill's value is different: it guarantees that the same fifteen analytical tests are applied in the same sequence with the same vocabulary, every time, regardless of project sector.

This has three consequences. First, gaps that fall outside the reviewer's habitual focus cannot be skipped; the framework forces completion across all fifteen rules. Second, findings are expressed in a standardised vocabulary, making cross-project comparison tractable in a way that prose summaries cannot be. Third, the output format is board-ready by design: a supervisory board member reviewing the report for a hydrogen project and the report for a hospital IT programme can compare maturity scores, flag counts, and omission vocabulary across both, without translating between

different analysts' frameworks.

3.3 Architecture: Two-File Structure

The skill is implemented as two files:

```
~/.claude/skills/systems-thinking/
+-- SKILL.md              -- workflow, output template, tone guidance
+-- references/
    +-- sys-rules.md      -- detailed criteria for all 15 SYS rules
```

The separation follows the Claude Code progressive disclosure principle: `SKILL.md` contains the audit workflow and exact output template (under 500 lines), while `sys-rules.md` is loaded on demand when applying the rules requires deeper criteria. This keeps the primary skill file within the context budget for all but the most resource-constrained deployments.

`SKILL.md` carries YAML frontmatter with a description that functions as the primary triggering mechanism. The description lists the specific phrase contexts that should activate the skill: "systems thinking audit", "check for systemic risks", "megaproject risk review", "SYS audit", "what could go wrong with this project", as well as implicit contexts such as a project director sharing a project document before a board approval gate.

3.4 The Five-Step Audit Workflow

The skill applies a defined five-step process:

1. **Frame the project** — establish type, phase, sector, decision gate, and information provided. Identify which of the fifteen rules are most likely to surface material gaps given the project characteristics.
2. **Apply all fifteen SYS rules** — rate each rule PRESENT, PARTIAL, or ABSENT based on the evidence provided. For PARTIAL and ABSENT ratings, write a finding that names the specific gap and its systemic consequence.
3. **Score the maturity level** — assign an overall maturity score on the 1–5 scale (see Section 3.6), calibrated to the distribution of PRESENT, PARTIAL, and ABSENT ratings and the severity of the highest-priority gaps.
4. **Rank critical gaps** — select the three to five gaps posing the greatest systemic risk, ordered by the combination of impact severity and reversibility. A gap that is remediable at pre-FID cost but catastrophic if discovered post-FID ranks above a gap that is material but discoverable and correctable during execution.
5. **Recommend next actions** — map gaps to tools from the minimal stack (see Section 3.8) and to the three integrated marketplace skills, and produce a prioritised action table with owner, output, and timeframe.

3.5 The Fifteen SYS Rules

The fifteen rules are derived directly from the failure patterns in the fifteen case studies. Each rule addresses a recurring failure class, states its rationale in one sentence, and specifies the PRESENT, PARTIAL, and ABSENT criteria that allow consistent rating across reviewers and projects. Appendix B lists the rules; full PRESENT/PARTIAL/ABSENT criteria are in `systems-thinking/references/sys-rules.md`. The rules are:

Rule	Short title	Core failure addressed
SYS-01	Full value chain	Missing upstream/downstream dependencies
SYS-02	Second and third-order effects	Indirect systemic consequences
SYS-03	Regulatory map before FID	Late legal and levy discoveries
SYS-04	Design-maturity threshold	Build-before-design-freeze
SYS-05	Interface ownership and evidence	Unmanaged system interfaces
SYS-06	Risk-adjusted estimate	False certainty in point budgets
SYS-07	Outside view first	Systematic optimism bias
SYS-08	Integration as its own megaproject	Late and under-resourced integration
SYS-09	Supply-chain and workforce readiness	FOAK supply-chain brittleness
SYS-10	Stakeholders as system components	Social licence as project risk
SYS-11	Business case stress test	Fragile economics under policy/price scenarios
SYS-12	Explicit kill criteria	Sunk-cost traps in prestige projects
SYS-13	Operations and maintenance in definition	ORAT and handover gaps
SYS-14	FOAK as learning programme	First-of-a-kind treated as serial production
SYS-15	Cluster and platform level	Sub-optimal standalone asset solutions

The rules are not a checklist to be completed mechanically. They constitute a system: SYS-04 (design maturity) and SYS-09 (supply-chain readiness) compound in FOAK programmes, as Olkiluoto 3 and Vogtle showed. SYS-03 (regulatory map) and SYS-11 (business case stress test) compound in market-dependent projects such as CCS or energy storage, where the regulatory framework directly determines the economics. Reviewers are instructed to identify and surface these compounding interactions rather than score each rule in isolation.

Note also the direct connection to the execution-phase skills: SYS-06 (risk-adjusted estimate) is operationalised by the `budget-estimator` skill (Section 4), and the interface register, kill criteria, and regulatory map produced in response to a systems-thinking audit become monitored artifacts in the execution-phase data store (Section 2.3).

3.6 Maturity Scoring

The maturity score aggregates the rule ratings into a single governance-facing signal:

Score	Description	Typical profile
1	Asset-centric only	Full chain absent; no regulatory map; interfaces undefined; integration not planned
2	Partial chain with major gaps	Chain sketched but incomplete; PARTIAL on most rules; at least one ABSENT on critical rules (SYS-03, SYS-06, SYS-07)
3	Structured approach, incomplete evidence	Framework present; key elements PARTIAL; integration and ORAT underweighted
4	Mature definition with minor gaps	Most rules PRESENT; remaining PARTIAL items are manageable; outside view applied

Score	Description	Typical profile
5	Fully integrated systems perspective	All rules PRESENT; chain, interfaces, regulation, operations, and exit logic complete

Most real projects reviewed to date score 2–3. A score of 4 or above at pre-FID is exceptional and merits verification that it reflects genuine evidence rather than optimism.

3.7 The Omission Vocabulary

Twelve standardised omission flags are embedded in the output template. Their purpose is to make gap identification directly comparable across projects and reviewers, replacing ad hoc language with a controlled vocabulary:

1. Full value chain not modelled
2. Regulatory due diligence missing
3. Interface owner unknown
4. Integration proof absent
5. Outside view missing
6. Kill criteria undefined
7. Design maturity not validated
8. Supply-chain readiness untested
9. Business case not stress-tested
10. FOAK not treated as learning programme
11. Stakeholder legitimacy not mapped
12. O&M not in scope of definition

Where a flag applies, the skill is instructed to use the exact phrase, not a paraphrase, in the Omission Flags section of the report. This enables automated screening of audit outputs across a project portfolio and makes it straightforward to track whether identified gaps are resolved at subsequent gate reviews. Appendix C maps each flag to its triggering rule rating.

3.8 The Minimal Tool Stack and Marketplace Integration

The research identifies a minimal stack adequate for early project definition review. The skill maps each stack element to its corresponding SYS rules:

Tool	SYS rules	What it delivers
Value-Chain Map	SYS-01, SYS-02	End-to-end dependency visibility
Regulatory Map	SYS-03	Permits, levies, liability, subsidies, tax treatment
Interface Register	SYS-05	Interface ownership and maturity evidence
ConOps / Operating Model	SYS-13	Day-in-the-life operational scenario
Scenario Stress Test	SYS-11	Business case at policy and price extremes
Monte Carlo Cost/Schedule	SYS-06	P50/P80/P90 risk-adjusted estimates
Reference Class Forecast	SYS-07	Outside view on cost and schedule
Supply-Chain Readiness Assessment	SYS-09	Top-10 supplier and craft labour gaps
Independent Red Team / Gate Challenge	SYS-12	External counter-review with kill criteria

Three existing Claude Code marketplace skills are integrated as recommended follow-on tools, matching specific SYS rules to purpose-built capabilities:

Skill	SYS rule(s)	Use case
what-if-oracle	SYS-11	Scenario stress: carbon ±50%, demand -20%, rates +200bp
perplexity-search	SYS-03	Regulatory due diligence for jurisdiction
literature-review	SYS-07	Reference class forecasting, IPA benchmarks

The skill does not auto-invoke these tools; it flags their applicability in the recommended next actions section, which preserves user control over which follow-on analyses to commission. These three tools address the three rules most commonly found ABSENT or PARTIAL in initial audits. Applying them in sequence converts a "maturity 2" project definition into a defensible stage-gate submission.

---

4. The Budget Estimator: Probabilistic CAPEX at Pre-FID Stage

4.1 Why Single-Point Estimates Fail Before Final Investment Decision

The standard practice in capital project development is to produce a single-point cost estimate at each stage gate: a number that is then treated as a commitment rather than a probability. This creates a structural problem: single-point estimates carry implicit assumptions about scope, productivity, and market conditions that are never made explicit and that are rarely interrogated by reviewers who see only the final number.

The consequence is systematic optimism bias. IPA research across 20,000 projects shows that single-point estimates systematically underestimate final cost because estimators anchor to base conditions and underweight the upper tail of the cost distribution. Tail events such as changed ground conditions, regulatory delays, and supply chain disruptions are individually unlikely but collectively almost certain to affect a multi-year capital project.

The AACE International recommended practice RP 18R-97 addresses this directly: a project's cost estimate should be accompanied by a probability distribution rather than a central value alone, and contingency should be sized to cover the P80 or P90 outcome, not the P50 alone. This is precisely the requirement that SYS-06 (risk-adjusted estimate) tests for in the systems-thinking audit. In practice, the requirement is rarely met because building a proper Monte Carlo model requires specialist software and significant effort. The budget-estimator skill removes that barrier.

4.2 What the Budget Estimator Delivers

Given a YAML work-package definition or a set of CLI parameters, the budget estimator runs 10,000 Monte Carlo iterations using triangular distributions calibrated to the selected AACE estimate class, and produces in under one minute:

- **P10, P50, and P90 CAPEX values** — the 10th, 50th, and 90th percentiles of the simulated cost distribution. P50 is the planning basis; P90 is the risk-adjusted ceiling for budget approval and contingency sizing

- **A cost distribution histogram** showing the full shape of the simulation output, which makes visible whether the distribution is approximately symmetric or heavily right-skewed (large upside risk)
- **A sensitivity tornado chart** ranking work packages by their Spearman rank correlation with total CAPEX across all iterations, which identifies precisely which cost elements drive the P90 and where additional engineering definition would most reduce uncertainty
- **A work-package breakdown chart** comparing P50 and P90 by WBS element, making it possible to direct contingency toward the packages that need it rather than spreading it uniformly

Each work package can be specified parametrically, using the heuristics library (EUR/km by pipe diameter and terrain, H2 material premium, compression station cost per MW), or as a direct cost entry with an uncertainty range. The AACE estimate class (1 through 5) is selected by the user and determines the default accuracy bounds applied to each work package: Class 5 (conceptual screening) applies -20%/+50%; Class 3 (FEED-stage study) applies -10%/+20%; Class 1 (definitive) applies -3%/+10%.

Contingency is reported as P90 minus P50, not as a flat percentage added to a point estimate. This means the contingency is sensitive to actual scope definition: packages with high parametric uncertainty contribute more to the P90 gap than packages with tight engineering definitions, which is the correct behaviour for managing pre-FID risk.

On the hydrogen pipeline example, the Class 3 estimate produces a P50 of EUR 1,833M and a P90 of EUR 1,930M against the approved FID CAPEX of EUR 1,749M. The P50 is 5 percent above the approved budget, within the $\pm 10/20\%$ accuracy band of a FEED-stage estimate, and the P90 implies a contingency requirement of EUR 97M (5.3%). The tornado chart identifies WBS-1.1 (Mainline North) as the dominant driver of the P90 spread, consistent with the ground conditions risk identified in the project risk register.

---

5. The EVM Module: From Calculation to Institutional Discipline

5.1 What Earned Value Management Is, and Why It Is Underused

Earned Value Management is the internationally recognised standard (ANSI/EIA-748) for integrating cost and schedule performance measurement. Its core logic is simple: at any point in a project, you can compute both what you have spent (Actual Cost, ACWP) and what you *should* have spent for the work you have actually completed (Earned Value, BCWP). The ratio of earned value to actual cost is the Cost Performance Index (CPI). A CPI of 0.91 means you are spending EUR 1.10 to deliver EUR 1.00 of budgeted work.

The power of EVM lies in its predictive validity. Research across thousands of projects has established that the CPI at the 20 percent completion milestone is a reliable predictor of final outcome. Projects that are underperforming at 20 percent completion rarely recover to budget; when they do, it is because management intervened early, not because efficiency spontaneously improved.[16]

Despite this, EVM is performed inconsistently in practice. The calculation requires integrating cost actuals, progress measurements, and the original budget baseline, data that typically live in three or four separate systems and require manual reconciliation. Under schedule pressure, this reconciliation gets abbreviated. The result is that the single most powerful early warning indicator

available to project management is produced late, inconsistently, or not at all.

5.2 What the EVM Module Delivers

Given a CSV file containing WBS codes, budget at completion (BAC), planned value (BCWS), earned value (BCWP), and actual cost (ACWP), all data that any functioning cost control system produces, the EVM module generates the following in under two minutes:

- **A complete markdown report** with executive summary, WBS-level performance table with RAG status, three EAC forecasts (CPI method, composite CPI×SPI method, and planned-rate method), variance at completion, and TCPI
- **Five publication-ready charts**: S-curve (BCWS/BCWP/ACWP over time), CPI/SPI trend with threshold bands, WBS cost variance waterfall, EAC forecast comparison, and a CPI/SPI quadrant map with bubble sizes proportional to budget at stake
- **Automatic escalation flags** when CPI or SPI breach thresholds, when SPI has been below 0.85 for two consecutive periods, or when TCPI exceeds 1.10 (the level at which recovery is generally considered unrealistic without re-baselining)
- **A plain-language interpretation** identifying the worst-performing package, the recommended EAC to use for board reporting, and the next three actions with named responsible parties

The RAG thresholds applied are industry-standard: CPI or SPI ≥ 0.95 = GREEN; 0.85–0.94 = AMBER; < 0.85 = RED. The composite CPI×SPI EAC method is recommended for board reporting, because it accounts for schedule pressure compounding cost efficiency loss, which is the dominant failure pattern in large capital projects.

For the hydrogen pipeline example included with the system, this method produces an EAC of EUR 1.637 billion against an approved BAC of EUR 1.350 billion for the contracted scope: a 21 percent overrun signal at month five of a 36-month execution programme, early enough for effective intervention. WBS-1.1 (Mainline North) is identified as the primary driver: its SPI of 0.750 (RED) reflects a linepipe delivery delay caused by port congestion, and if SPI remains below 0.85 in the June reporting period, the two-consecutive-period escalation rule fires automatically, generating a mandatory written notice to the Project Director.

5.3 Integration with Existing Tools

The system reads data that capital project organisations are already producing; it does not require replacing any existing tool.

Primavera P6. Schedule data is ingested via `.xer` export files using the P6XER MCP (Model Context Protocol) server. This allows Claude Code to read P6 project schedules directly, extract critical path float, identify milestone forecast dates, and feed schedule data into the schedule-analyzer agent, without requiring a P6 licence or database connection in the AI environment.

Excel and CSV cost systems. Any cost control system that can export EVM data in tabular form (BCWS, BCWP, ACWP by WBS code) is compatible. The EVM skill accepts both snapshot and time-phased formats and validates column names on intake.

Document repositories. Contract PDFs, engineering specifications, and tender documents are processed via the `pdf` skill. The contract-reviewer agent combines PDF extraction with structured templates to produce commercial summaries from documents that would otherwise require hours of manual review.

Reporting platforms. The `pptx` skill converts the monthly EVM markdown report and the five generated chart PNGs into a presentation-ready deck, formatted to the organisation's template. This output feeds directly into the monthly board reporting pack without manual transcription.

Atlassian and collaboration tools. The Atlassian MCP server enables action items generated by the AI (e.g., from the risk-assessor or schedule-analyzer agents) to be logged directly as Jira tickets with assigned owners and due dates, which closes the loop from AI analysis to trackable task.

---

6. Evaluation: Does the Skill Approach Work?

The claim underlying the entire suite is that encoding analytical frameworks as skills produces more consistent, complete, and comparable outputs than relying on unaided AI capability. It has been tested empirically on the `systems-thinking` skill, the component where the claim is least obvious. For the EVM and budget-estimator skills, the computational core is deterministic Python; consistency is a property of the code. For the behavioural skills (the audit skill, and by extension `gate-readiness` and `lessons-learned`, which enforce checklists and templates the same way), consistency must be demonstrated.

6.1 Methodology

The skill was evaluated using the `skill-creator` framework, which runs parallel Claude Code agent instances, one with the skill loaded and one without, against the same test prompts, then grades outputs against structured assertions.

Three test scenarios were designed to cover different project types, decision gates, and audience profiles. For each scenario, both configurations produced a full audit report. The reports were graded against assertions verifiable from the text. Timing and token data were recorded at completion.

The evaluation measures what the skill adds relative to an unaided capable Claude model, not the absolute quality of either output.

6.2 Test Scenarios

Scenario 1: Offshore Wind and Green H2, Norway (pre-FID)

A 500 MW offshore wind installation combined with a hydrogen electrolysis facility, targeting green hydrogen export via a proposed TSO pipeline network. The scenario provides a project brief including installed capacity targets and high-level CAPEX figures. The decision gate is FID; the audience is the project steering committee.

The systemic risks in this scenario include an unmodelled hydrogen export chain (SYS-01), early-stage TSO pipeline planning (SYS-03), and a business case that has not been stress-tested against carbon price and demand fluctuations (SYS-11).

Scenario 2: Hospital EHR Transformation, Germany

A €340 million EHR transformation programme spanning eight German hospitals, with Phase 1 declared complete but Phase 2 (clinical integration and workflow adoption) not started. The scenario provides a programme update presented to the supervisory board.

The systemic risks in this scenario include the critical distinction between software installation and clinical integration (SYS-08), undefined criteria for continuation or termination given sunk costs (SYS-12), and an operating model that has not been validated in clinical workflow terms (SYS-13).

Scenario 3: FOAK LiOH Refinery, Finland

A €1.1 billion first-of-a-kind LiOH refinery project based on Australian spodumene feedstock, projecting an 18% IRR at prevailing lithium prices. The scenario provides a stage-gate pack for a PMO review. The project has not yet obtained Finnish environmental permits.

The systemic risks in this scenario include an unverified IRR in a commodity market that lost 80% of its value over 24 months (SYS-11), missing YVA environmental permit in Finland (SYS-03), FOAK technology risk treated as a standard EPC delivery problem (SYS-14), and a single-source spodumene feedstock from one Australian mine (SYS-09).

6.3 Assertions

Assertions were designed to test structural and vocabulary compliance, not domain knowledge. Domain-knowledge assertions such as "does the output identify the hydrogen pipeline gap?" were expected to pass in both configurations, because an unaided Claude model has adequate knowledge of hydrogen value chains, Finnish permitting law, and lithium market dynamics. The discriminating assertions test whether the skill-mandated framework is applied:

- **contains_maturity_score** — output contains an explicit X/5 score
- **covers_all_15_rules** — output contains SYS-01 through SYS-15 with ratings
- **uses_rating_vocabulary** — PRESENT / PARTIAL / ABSENT used throughout
- **uses_omission_vocabulary** — at least one exact omission flag from the twelve-item vocabulary
- **identifies_critical_gaps** — ranked gap section with at least three gaps
- **recommends_tool_stack** — specific named tools recommended in next actions

Domain-specific assertions varied by scenario:

```
flags_hydrogen_pipeline_gap
flags_integration_gap
flags_kill_criteria
flags_foak_risk
flags_permit_gap
flags_business_case_stress
```

6.4 Results

Configuration	Pass rate	Mean time	Mean tokens
With skill	**100%**	190.6 s	26,805
Without skill	41%	335.1 s	24,586
Delta	**+59 pp**	-144.5 s	+2,219

The 59-percentage-point pass rate delta is driven entirely by the structural and vocabulary assertions. Without the skill:

- Scenario 1 invented an eight-dimensional framework; no SYS-01 to SYS-15 labels appeared; omission vocabulary was absent; the tool-stack recommendation was generic

- Scenario 2 used eight "systems thinking" dimensions (stocks, flows, feedback loops); integration was discussed but the installed-vs-integrated distinction was not named with the skill's vocabulary; kill criteria were addressed only in general terms
- Scenario 3 used ten categorical dimensions with CRITICAL/HIGH/MEDIUM severity ratings; FOAK risk was discussed without SYS-14; no exact omission flags appeared

In all three scenarios, the without-skill output correctly identified the material domain risks. The score penalty was structural, not substantive. This is the value proposition: the skill does not supply knowledge that an experienced reviewer lacks. It supplies the framework that makes outputs comparable across projects and auditable across gate reviews.

The with-skill outputs were also faster by an average of 144.5 seconds. The structured fifteen-rule framework reduced decision overhead: instead of constructing an analytical framework de novo, the agent applied a defined one. The small token premium (+2,219) reflects the additional structure loaded from the skill file.

6.5 Strongly and Weakly Discriminating Assertions

A post-hoc analysis of assertion behaviour identified two categories:

Strongly discriminating (skill 100%, baseline 0%): All structural and vocabulary assertions:

```
covers_all_15_rules
uses_rating_vocabulary
uses_omission_vocabulary
recommends_tool_stack
```

These assertions fail in every without-skill run because the unaided model invents a different framework each time.

Non-discriminating (both pass): Domain-knowledge assertions: whether the output identifies the hydrogen pipeline gap, the Finnish permit requirement, the 18% IRR stress-test problem, the spodumene supply-chain risk. Both configurations pass these in all three scenarios. The conclusion is confirmed: the skill does not supply domain knowledge; it supplies structural discipline.

The generalisation to the execution-phase components follows directly: the value of encoding escalation thresholds, output templates, and heuristic provenance rules in `CLAUDE.md` and `SKILL.md` files is the same structural discipline, applied to the monthly controls cycle instead of the FID gate. Appendix D contains the per-scenario evaluation detail.

---

7. Investment Case, Risks, and Governance

7.1 The Value Drivers

The business case for AI-augmented project management operates on four distinct value levers:

Lever 1 — Systemic gaps surfaced before capital is committed

The pre-FID window is where remediation is cheapest. Every one of the fifteen failure cases in Section 1.2 involved a gap that was knowable before FID (and in most cases known to someone), but that was not surfaced to the decision gate in a form the approving body could act on. A structured audit that forces completion across all fifteen rules, at a cost of hours rather than weeks,

changes the economics of gate review: the board sees a maturity score, a ranked gap list, and an explicit statement of what has *not* been evidenced, before committing capital. A single regulatory or value-chain gap caught pre-FID, where the remedy is analysis and renegotiation rather than write-off, can be worth the entire programme cost many times over.

Lever 2 — Earlier detection of adverse trends in execution

Research by IPA establishes that the cost of corrective action increases approximately exponentially with the delay between signal and response. A schedule slip that can be corrected for EUR 2 million in month three may require EUR 15–20 million to recover by month nine, because interim procurement commitments, subcontractor mobilisation costs, and lost float compound the original deviation.

The system makes the trend visible in the first reporting period it appears, rather than the third. On a EUR 1 billion project, even one early intervention that prevents a 2 percent cost growth generates a return that exceeds the entire cost of implementing and operating the system.

Lever 3 — Consistent application of escalation rules

Escalation failures are documented as a primary driver of late-stage project crises. The mechanism is well understood: a cost engineer identifies a negative trend, judges it borderline, decides to wait one more period to confirm before escalating, and the window for low-cost intervention closes. Encoded escalation thresholds eliminate the discretion at the margin. When CPI falls below 0.85, the flag fires automatically. The Project Director is notified regardless of where the cost engineer's judgment falls.

Lever 4 — Reduced cost of producing controls outputs

Companies using AI-assisted project management tools report an average 15 percent improvement in project delivery productivity.[17] Monthly reporting typically consumes 3–5 working days per reporting period for a major project's controls team; consistent findings indicate that AI assistance reduces this by 40–60 percent. This frees experienced project controls professionals to perform root cause analysis, contractor engagement, and recovery planning rather than data assembly and chart production.

7.2 Conservative Financial Model

The following model is intentionally conservative and uses a single EUR 1 billion project as the unit of analysis. It quantifies only the execution-phase levers; the pre-FID audit value (Lever 1) is treated as upside because its realisation depends on whether a material gap exists to be caught:

Value Driver	Basis	Annual Value (EUR)
Earlier detection — 1 avoided 2% cost growth	One intervention per project year	20,000,000
Reporting efficiency — 2 FTE-months freed per year	Senior engineer at EUR 15k/month fully loaded	360,000
Consistent escalation — avoided one late crisis	50% probability × EUR 5M average cost of late recovery	2,500,000
Total (conservative)		**~22,860,000**

Against this, the implementation costs are modest:

Cost Item	Basis	Annual Cost (EUR)
Claude Code enterprise licences (10 users)	Current enterprise pricing	~60,000
Internal implementation effort (one-time)	15 person-days × senior engineer rate	~30,000
Ongoing maintenance and data governance	0.25 FTE	~75,000
Total		**~165,000 p.a.**

The implied return on investment exceeds 100:1 on conservative assumptions, driven almost entirely by the value of a single avoided late-stage intervention. The ratio improves further on a portfolio of projects, where the fixed infrastructure cost (licences, governance, templates) is shared, and where the systems-thinking audit adds portfolio-level comparability that no amount of ad hoc review can provide.

7.3 Contextual Benchmark

Companies that use AI-driven tools in project management deliver 61 percent of their projects on time, compared to 47 percent for those that do not, a 14-percentage-point improvement.[18] On a programme of five concurrent capital projects, each averaging EUR 500 million in CAPEX, moving one project from the "late" to "on-time" bucket, with a typical delay cost of 5–10 percent of CAPEX, represents EUR 25–50 million in value creation from the portfolio uplift alone.

The open-source code base eliminates implementation risk as a financial objection: the system can be inspected in detail before deployment, extended without vendor dependency, and discontinued without sunk cost if the pilot does not demonstrate value.

7.4 AI-Specific Risks

Risk: AI produces confident but incorrect analysis

Likelihood: Medium. Consequence: Medium. The system mitigates this through schema validation, provenance requirements on all heuristics, and mandatory cross-checking of EAC against parametric benchmarks. For the audit skill, the rating criteria (PRESENT / PARTIAL / ABSENT) are evidence-based: absence of evidence is rated ABSENT, never inferred as adequate. No AI output is presented as final without a named human reviewer. The system generates recommendations; it does not make decisions.

Risk: Sensitive commercial data processed through third-party AI systems

Likelihood: Low with controls. Consequence: High. Claude Code can be deployed on-premises or in a private cloud environment using Anthropic's enterprise API. No project data need leave the organisation's controlled infrastructure. The data governance policy establishes which data categories are permissible and in what environment.

Risk: Over-reliance reduces human expertise over time

Likelihood: Low with design. Consequence: Medium. This risk is real and documented in analogous automation contexts. The mitigation is design: the system explicitly requires human interpretation

of every report, ensures that escalation recommendations are reviewed not auto-executed, and generates plain-language explanations that build rather than bypass analytical understanding.

7.5 Organisational Risks

Risk: Resistance from project controls professionals

Likelihood: High without change management. Consequence: Medium. The most effective mitigation is framing. This system removes the least engaging portions of a cost engineer's work (data assembly, chart production, threshold checking) and returns time for the high-judgment work that experienced professionals value. Early engagement of senior project controls staff in the pilot design, and visible credit for the improved reporting outputs, is the primary change management lever.

Risk: Audit findings treated as box-ticking

Likelihood: Medium. Consequence: Medium. A fifteen-rule framework can degenerate into compliance theatre if gate bodies accept ratings without interrogating the underlying evidence. The mitigation is the omission vocabulary and gate tracking: flags raised at one gate review must be explicitly closed or accepted at the next, and a maturity score of 4+ at pre-FID triggers verification rather than congratulation.

Risk: Data quality too poor for meaningful analysis

Likelihood: Medium. Consequence: Medium. The system will surface data quality problems that were previously obscured by manual report production. This is a feature, not a bug. A data quality review before the system goes live on the pilot project establishes a clean baseline and closes the gaps that would otherwise persist undetected.

Risk: Pilot succeeds but rollout stalls

Likelihood: Medium without board sponsorship. Consequence: High. A successful pilot followed by a stalled rollout is the dominant failure mode for enterprise technology adoption. The mitigation is board-level ownership of the rollout mandate and a defined CAPEX threshold above which the system is mandatory, not optional.

7.6 What This System Does Not Do

Equally important is what this system does not replace:

- It does not replace the Project Director's judgment on whether to escalate a situation to the board, or the board's judgment on whether to approve FID
- It does not supply domain knowledge that an experienced reviewer lacks; it supplies the framework that makes review complete and comparable
- It does not negotiate with contractors, assess claim merits, or draft legal correspondence without human review
- It does not produce FID-quality bottom-up cost estimates; the budget estimator produces parametric estimates with explicit AACE-class accuracy bounds, and the EVM module analyses cost *performance* against an already-established baseline
- It does not replace the physical site inspection, quality review, or HSE incident investigation

- It does not guarantee project success; it improves the information available to the humans who determine outcome

Projects succeed because of skilled, experienced, well-led teams. This system makes it harder for those teams to miss systemic gaps at the definition stage, harder to miss early warning signals in execution, easier to produce consistent reporting, and more likely that the right information reaches the right decision-maker at the right time.

---

8. Usage Guide

8.1 Installation

All components install by copying directories into the Claude Code skills folder and, for the Python-based skills, running one `pip\install`:

```
SK=~/.claude/skills
cp -r evm/ $SK/evm/
cp -r budget-estimator/ $SK/budget-estimator/
cp -r systems-thinking/ $SK/systems-thinking/
cp -r gate-readiness/ $SK/gate-readiness/
cp -r lessons-learned/ $SK/lessons-learned/
pip install -r $SK/evm/requirements.txt
pip install -r $SK/budget-estimator/requirements.txt

# Project scaffolding (per project)
cp scaffolding/CLAUDE.md my-project/CLAUDE.md
cp -r scaffolding/.claude/ my-project/.claude/
cp -r scaffolding/heuristics/ my-project/heuristics/
cp -r scaffolding/schemas/ my-project/schemas/
cp scaffolding/templates/project-brief.md \
  my-project/project-brief.md
```

Claude Code discovers the skills at startup; no further configuration is required.

8.2 Invoking the Skills

Each skill triggers automatically on intent, not exact keywords:

Skill	Trigger examples
`systems-thinking`	"systems thinking audit", "SYS audit", blind-spot check; project docs before board/FID gate
`budget-estimator`	"estimate CAPEX", "P50/P90", Monte Carlo; YAML with `work_packages:`
`evm`	"run EVM", "earned value", CPI/SPI, EAC; CSV with bac/bcws/bcwp/acwp columns
`gate-readiness`	"gate readiness", "are we ready for FID?", "assemble the gate pack", stage-gate submission
`lessons-learned`	"lessons learned", "close-out review", "estimate vs actual", "calibrate heuristics"

8.3 What Input to Provide to the Audit Skill

The systems-thinking skill can work from any project documentation available. The richer the input, the more specific the findings. Useful inputs include:

- **Project brief or definition** — scope, CAPEX, timeline, technology, location, decision gate
- **Business case or investment paper** — financial model, IRR assumptions, market context
- **Programme update or status report** — progress against plan, issues log, next milestones
- **FID or stage-gate pack** — technical feasibility, regulatory status, contracting strategy, risk register

Where specific information is unavailable, the skill rates the relevant rule ABSENT and flags the omission. Absence of evidence is itself a finding: if the regulatory map has not been produced, the project does not have a regulatory map, and SYS-03 is ABSENT regardless of whether the regulatory risk is subsequently manageable.

8.4 Reading the Audit Output

The audit report has a defined structure:

```
## Project Overview
[Project name, type, phase, decision gate, audience]

## Overall Maturity Score: X/5
[One-sentence characterisation of the maturity level]

## SYS Rule Assessment

| Rule | Rating | Finding |
| --- | --- | --- |
| SYS-01 | PRESENT / PARTIAL / ABSENT | [Specific finding or confirmation] |
...

## Critical Gaps (ranked)
[Top 3-5 gaps, ordered by systemic impact and reversibility, with brief rationale]

## Recommended Next Actions
[Table of gap → tool → owner → output → timeframe]

## Omission Flags
[Exact flags from the twelve-item vocabulary that apply to this project]

## Board Guidance Note
[Optional: plain-language summary for a non-technical board audience]
```

The overall maturity score is the governance-facing signal: it summarises the distribution of PRESENT, PARTIAL, and ABSENT ratings into a single comparable metric. The critical gaps section is the action-facing signal: it tells the project team where to direct effort before the next decision gate. The omission flags are the audit-facing signal: they produce a machine-readable list that can be tracked across gate reviews to verify closure.

8.5 The Monthly Execution Workflow

1. **At reporting cutoff** — append a new period row for each WBS element to `03-cost/evm-timephased.csv`
2. **Run EVM** — open Claude Code in the project directory and type *"Generate the monthly EVM report"*. Claude finds the data, runs the skill, displays the report and charts, and highlights any escalation flags
3. **Update milestones** — update `02-schedule/milestones.csv` with new forecast dates; type *"schedule analysis"* to invoke the schedule-analyzer agent

4. **Review risks** — type *"risk review"* to invoke the risk-assessor agent; it identifies new risks needing entry and flags materialised risks for cost register update
5. **Assemble report deck** — use the `pptx` skill to build the monthly progress presentation from the EVM report and chart files
6. **At decision gates** — type *"gate readiness"* to assemble and assess the evidence pack before any stage-gate or re-baselining decision
7. **At package or project completion** — type *"lessons learned"* to run the close-out review and generate the heuristics calibration proposal

---

Appendix A — Key Metrics Produced by the EVM Module

Metric	Formula	What it tells management
CPI	BCWP ÷ ACWP	Cost efficiency: EUR earned per EUR spent
SPI	BCWP ÷ BCWS	Schedule efficiency: earned value vs plan
EAC (Composite)	ACWP + (BAC-BCWP) ÷ (CPI×SPI)	Final cost forecast, accounting for schedule pressure
VAC	BAC - EAC	Expected over- or under-run at completion
TCPI	(BAC-BCWP) ÷ (BAC-ACWP)	CPI required for all remaining work to finish within budget; >1.10 indicates re-baselining is needed

RAG thresholds (industry standard)

Index	GREEN	AMBER	RED
CPI	≥ 0.95	0.85 – 0.94	< 0.85
SPI	≥ 0.95	0.85 – 0.94	< 0.85

Appendix B — The Fifteen SYS Rules

Full PRESENT, PARTIAL, and ABSENT rating criteria for each rule are in `systems-thinking/references/sys-rules.md` in the repository. Summary:

Rule	Short title	Core failure addressed
SYS-01	Full value chain	Missing upstream/downstream dependencies
SYS-02	Second and third-order effects	Indirect systemic consequences
SYS-03	Regulatory map before FID	Late legal and levy discoveries
SYS-04	Design-maturity threshold	Build-before-design-freeze
SYS-05	Interface ownership and evidence	Unmanaged system interfaces
SYS-06	Risk-adjusted estimate	False certainty in point budgets
SYS-07	Outside view first	Systematic optimism bias
SYS-08	Integration as its own megaproject	Late and under-resourced integration
SYS-09	Supply-chain and workforce readiness	FOAK supply-chain brittleness
SYS-10	Stakeholders as system components	Social licence as project risk

Rule	Short title	Core failure addressed
SYS-11	Business case stress test	Fragile economics under policy/price scenarios
SYS-12	Explicit kill criteria	Sunk-cost traps in prestige projects
SYS-13	Operations and maintenance in definition	ORAT and handover gaps
SYS-14	FOAK as learning programme	First-of-a-kind treated as serial production
SYS-15	Cluster and platform level	Sub-optimal standalone asset solutions

---

Appendix C — Omission Vocabulary

The twelve standardised flags are used verbatim in all audit outputs. Their purpose is to produce a machine-readable, comparable signal across projects and gate reviews:

Flag	Triggered when
Full value chain not modelled	SYS-01 ABSENT or PARTIAL
Regulatory due diligence missing	SYS-03 ABSENT
Interface owner unknown	SYS-05 ABSENT or PARTIAL (no named owner)
Integration proof absent	SYS-08 ABSENT
Outside view missing	SYS-07 ABSENT
Kill criteria undefined	SYS-12 ABSENT
Design maturity not validated	SYS-04 ABSENT
Supply-chain readiness untested	SYS-09 ABSENT
Business case not stress-tested	SYS-11 ABSENT
FOAK not treated as learning programme	SYS-14 ABSENT (FOAK projects only)
Stakeholder legitimacy not mapped	SYS-10 ABSENT
O&M not in scope of definition	SYS-13 ABSENT

Appendix D — Evaluation Benchmark Detail

Eval	Configuration	Pass rate	Passes / Total	Time (s)	Tokens
Offshore wind + hydrogen	With skill	100%	8/8	181.0	26,768
Offshore wind + hydrogen	Without skill	38%	3/8	447.9	30,026
Hospital EHR transformation	With skill	100%	7/7	178.7	26,610
Hospital EHR transformation	Without skill	43%	3/7	267.9	21,559
FOAK lithium refinery	With skill	100%	7/7	212.2	27,038
FOAK lithium refinery	Without skill	43%	3/7	289.4	22,172
Aggregate with skill		**100%**	**22/22**	**190.6**	**26,805**
Aggregate without skill		**41%**	**9/22**	**335.1**	**24,586**

Appendix E — Technical Resource

The working system (EVM skill, budget estimator, systems-thinking audit skill, scaffolding templates, domain heuristics, JSON schemas, specialist agent definitions, and a complete example project) is available at:

github.com/jmeier1963/large_capital_project_management

The repository includes:

- `evm/evm_calculator.py` — EVM engine (713 lines, no dependencies beyond pandas and matplotlib)
- `budget-estimator/budget_estimator.py` — parametric CAPEX engine with P50/P90 Monte Carlo
- `systems-thinking/` — audit skill, SYS rule criteria, evaluation scenarios
- `gate-readiness/` — stage-gate evidence pack skill with per-gate evidence matrix
- `lessons-learned/` — close-out review and heuristics calibration skill
- `scaffolding/` — project configuration templates for immediate deployment
- `examples/H2-PIPE-DE-001/` — worked example: 500 km H2 pipeline, EVM data
- `README.md` — step-by-step installation and usage instructions

Installation requires copying skill directories and running one `pip\install` command per Python-based skill.

References

1. Flyvbjerg, B. & Gardner, D. (2023). *How Big Things Get Done*. Macmillan. Database of 16,000 projects; 8.5% delivered on cost and schedule; 0.5% delivered all promised benefits.
2. McKinsey & Company (2015). "Megaprojects: The good, the bad, and the better." Review of 300+ projects exceeding USD 1 billion. Average 80% cost overrun, 50% schedule delay.
3. McKinsey & Company (2017). "Increasing transparency in megaproject execution." Analysis of 48 troubled projects; 73% of overruns attributed to execution failures.
4. Flyvbjerg, B. (2017). "The Iron Law of Megaprojects." *Cato Institute Policy Report*. 91.5% of projects exceed budget, schedule, or both.
5. McKinsey Global Institute (2016). "Reinventing Construction." Mining and metals sector overrun data. Rail, bridge, and tunnel statistics from Flyvbjerg's database.
6. Hydrogen Council & McKinsey & Company (2025). *Global Hydrogen Compass*. USD 110B committed investment across 500+ post-FID projects.
7. Case studies and root-cause analysis drawn from public post-mortems including NAO reports on Crossrail (2019), carbon capture (2017), and NPfIT (2011); STUK on Olkiluoto 3 (2007); ASN on Flamanville 3; DOE on Vogtle; National Academies on the Big Dig; Holyrood Audit Committee (2004); Rio Tinto on Oyu Tolgoi; Trafikverket on Hallandsås; Barrick on Pascua-Lama; and Gassnova on Mongstad CCS.

8. NAO. (2017). *Carbon capture and storage: lessons from the competition*. London: National Audit Office. Primary source for Longannet full-chain economics and funding architecture failure.

9. STUK. (2007). *Summary Investigation Report: Olkiluoto 3*. Finnish Radiation and Nuclear Safety Authority. Design maturity, subcontractor control, and quality assurance findings.

10. NAO. (2019). *Crossrail: a progress update*. HC 2004, 2017–2019. London: National Audit Office. Integration planning, ORAT, and safety assurance underestimation findings.

11. Trafikverket-related documentation on Hallandsås Tunnel (1992–2015); Barrick Gold. (2012). Annual Report and investor presentations on Pascua-Lama.

12. Scottish Parliament Audit Committee. (2004). *Holyrood: The Management of the Holyrood Building Project*. Edinburgh: Scottish Parliament. Governance fragmentation and accountability diffusion findings.

13. Meier, J. (2026). *Systems Thinking in Large-Scale Projects: Research Report*. Unpublished research report. Internal reference for the research basis of the skill design. (See systems-thinking-research-report-en.md in the repository references folder.)

14. Independent Project Analysis (IPA). *Capital Project System Improvement*. Project Control Index methodology and performance correlation. ipaglobal.com

15. Deloitte US (2026). *State of AI in the Enterprise*. 89% of companies have deployed AI; 94% report not seeing significant value.

16. Christensen, D.S. (1993). "The Estimate at Completion Problem: A Review of Three Studies." *Project Management Journal*. CPI stability at 20% completion and its predictive validity for final outcome.

17. Various sources aggregated in: Celoxis (2025). "AI and Machine Learning in Project Management." 15% average productivity improvement; 61% on-time delivery with AI tools vs. 47% without.

18. Ibid.

Additional methodological sources: GAO (2020), *Cost Estimating and Assessment Guide*, GAO-20-195G — Monte Carlo, reference class, and estimate quality methodology; NAO (2013), *Over-optimism in government projects* — systematic optimism-bias evidence and outside-view correction rationale.