FAROS

FAROS RESEARCH · TECHNICAL WHITE PAPER

The Model Is Not the System

The case for choosing AI coding routes
on evidence from your own engineering work,
drawn from 2,532 route-task results across 211 real tasks

Faros Research

Method: Faros Time Machine
Version 1.0 · July 2026

EXECUTIVE SUMMARY

Two releases, four findings, one operating discipline.

Written for engineering leaders, platform teams, and AI program owners.

AI coding systems are commonly compared by model name. Engineering organizations deploy something larger: a model operating through a coding harness, provider, runner configuration, tools, instructions, and runtime controls on a particular task. That complete system produces the result. This paper makes that case: route decisions belong on controlled evidence from the organization's own historical work.

Within the Max-Effort release, mean projected cost per task ranged from $0.95 to $6.83, a 7.2× spread, and the highest-cost route was not the quality leader. Route selection is therefore an economic as well as a technical decision.

Faros built **Time Machine** to produce that evidence from completed engineering work. The **Faros Route Index** demonstrates the method on a fixed cohort of 211 anonymized historical tasks from 12 Faros repositories. Across two releases, 12 dated route configurations produced 2,532 route-task results through July 17, 2026.[1,3] Four findings stand out:

HISTORICAL TASKS	ROUTE-TASK RESULTS	DATED RELEASES	CONFIGURATIONS
211	**2,532**	**2**	**12**

01 **Open-model routes recorded the highest mean quality in both releases.** In the Original release the top three routes by mean quality all ran open models, and the strongest frontier-model route trailed the leader by 4.71 points. In Max-Effort, Claude Code + GLM-5.2 Fast led at 65.46%, with the strongest frontier-model route 15.43 points behind.

02 **The complete route changed the result even when the model label stayed the same.** Same-model route bundles differed by 5.22 points for Kimi K2.6 and 6.68 points for Opus 4.8 in the Original release, and by 17.48 points for GLM-5.2 Fast and 3.47 points for Kimi K2.7 Code in Max-Effort. Paired intervals separate three of the five pairings clearly; the other two include zero.

03 **Task-level evidence revealed information that portfolio averages compressed.** The Original release's top two routes were effectively tied at the portfolio level, 0.16 points apart, yet each led on a substantial set of the 211 paired tasks.

04 **The strongest quality result did not require the highest projected cost.** In both releases, the mean-quality leader cost less per task than another route with a lower quality score.

Together, these findings support a practical operating discipline: evaluate complete routes, use representative historical work, compare alternatives on the same tasks, examine quality and economics together, and refresh the evidence when the system or workload changes.

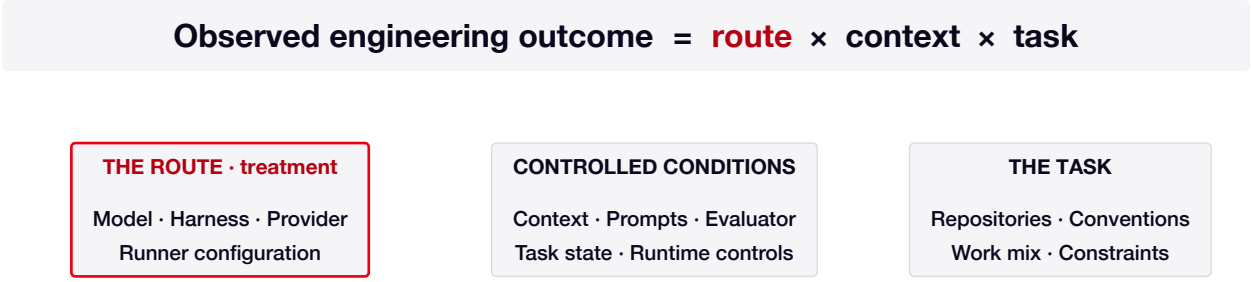

Exhibit 1 **Anatomy of an AI coding system.**

The route is the declared treatment; context, prompts, evaluator, task state, and runtime controls are the controlled conditions · Source: Faros Time Machine method

SECTION 1 · THE DEPLOYMENT DECISION

AI coding is a systems decision.

Foundation models have become the most visible layer of AI-assisted software development. Their names anchor benchmark tables, purchasing discussions, and release announcements. But a model does not reach a repository by itself.

Before an AI coding system can investigate a task or produce a patch, an organization has already made a series of technical choices: which harness plans, searches, edits, and recovers; which provider serves the model; which runner configuration sets budgets and limits; which instructions and repository context reach the agent; which tools and permissions the system can use; and which engineering task the system is attempting.

Faros calls the declared combination of model, harness, provider, and runner configuration a **route**. The deployed system combines that route with its context, tools, controls, and task.

The harness shapes how a model explores and modifies a codebase, while the provider and runner define the operating envelope. Context sets what the agent knows before it acts, and tools determine what it can inspect and verify. The task decides which of these capabilities matter. A model name therefore identifies one component of the deployment decision. The complete route, operating with its context, tools, and controls on the task, is the system that reaches the repository.

From general capability to deployment evidence

Public benchmarks provide valuable evidence about general model and system capability.[4] They make standardized comparisons possible and help organizations identify promising candidates.

The deployment decision requires an additional layer of evidence. An engineering organization needs to know how a complete route behaves on its repositories, conventions, task descriptions, architecture, and

mix of work. That question cannot be answered by transferring a model rank directly from an unrelated task population; evaluation guidance from NIST makes the same point, recommending that benchmarks be selected to fit the evaluation objective.[5]

The required unit of analysis is a complete route attempting representative work under a controlled comparison. The required evidence is paired at the task level so that alternative routes face the same engineering problems. Time Machine produces that evidence.

The deployment decision is a route choice under local constraints.

SECTION 2 · HOW TIME MACHINE MEASURES

Completed work becomes a controlled replay.

Time Machine is a controlled historical-simulation and measurement method. It turns completed engineering work into replayable comparisons of declared AI coding configurations. The method begins with a real work item and the accepted engineering change that resolved it. That history provides three essential elements: the original problem, the repository state before the solution, and evidence describing what successful resolution required.

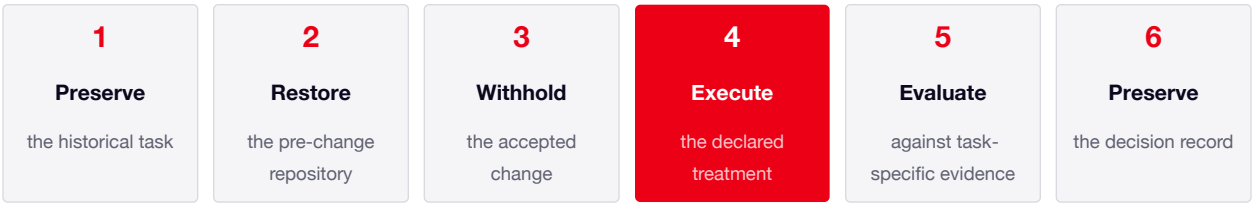

Exhibit 2 **Historical replay.**

Completed work becomes decision evidence; the candidate route never receives the accepted implementation · Source: Faros Time Machine method

The result is a controlled historical replay:

> *Session analytics describe what happened. Time Machine shows what each declared alternative produced when it attempted the same historical work under the declared comparison controls.*

Two controlled experiment types

	Route comparison	Context comparison
Decision	Which complete route performs best on this work?	Does a declared context change improve outcomes on this work?
Treatment	Model, harness, provider, and runner configuration	Effective scoped context supplied to the agent
Held constant	Task cohort, context, evaluation, and run controls	Task cohort, route, evaluation, and run controls
Evidence	Quality and economics for route selection	Quality evidence for a context decision

The Faros Route Index represents **route comparison**. Context comparison is a distinct treatment that applies the same historical-replay principle while holding the route fixed.

Time Machine and the Route Index

Time Machine creates the historical comparison. The Route Index publishes the result in a form that engineering leaders can inspect. This separation is important: **Time Machine** is the replay and evaluation method; **the Faros Route Index** is the dated public evidence surface; **a route** is the complete declared AI coding configuration being compared. Providers and models appear as components within routes. They are not the Time Machine architecture.

SECTION 3 · EVIDENCE BASE AND MEASUREMENT

A fixed complete-case cohort, observed twice.

The Route Index contains two releases evaluated on the same fixed cohort of 211 anonymized historical Faros engineering tasks.

Release	Date	Routes	Tasks / route	Results
Original	June 25, 2026	7	211	1,477
Max-Effort	July 17, 2026	5	211	1,055
Combined		**12**	**211 paired**	**2,532**

Every route attempted every task within its release. That paired design keeps the work constant while the route changes. A route average describes performance across the cohort; the underlying task-level matrix

shows how the aggregate was produced. The 211-task cohort is the strict subset of a larger selected pool of historical work; the selection funnel and validity gates are described in Methods. The Index records two decision measures for every route-task result: the **Agentic Rubric score**, a continuous, task-specific quality score from 0 to 100%, and **projected cost**, the estimated route cost associated with producing the result. Wherever this report says quality, it means the Agentic Rubric score; scoring is execution-free.

Measuring engineering quality

Software tasks can admit more than one sound implementation. They can also produce meaningful partial progress that a binary outcome does not describe. Time Machine uses an **Agentic Rubric** to represent the engineering requirements of each task as weighted, concrete criteria across four dimensions: file-change scope, alignment with the work specification, integrity of the implementation, and expected runtime behavior.[2]

The rubric is written for each task before any candidate solution is evaluated. It is a weighted checklist of the important properties of a correct solution, and its criteria are concrete questions that can be answered by inspecting the proposed change, such as which files should change and whether the root cause is addressed, rather than vague requirements like "the patch should be correct."[2] Every route compared on a task is scored against that same rubric.

A route-blinded judge receives exactly three inputs: the task description, the candidate change, and the rubric. The judge does not receive the identity of the model or route that produced the work, and it is not asked to assign an impressionistic rating. For each criterion it answers yes or no, and the resulting score is the weighted percentage of task criteria satisfied.[2]

Faros has evaluated this measurement framework against ground-truth software tasks:[2] accepted reference patches averaged 80%, with a median of 85%, on their own rubrics; repeated judgments showed typical variation of approximately ±2 points and test-retest correlation of approximately 0.96; known successes ranked above known failures with per-task AUC of 0.75–0.86 on validated repositories; and

ACCEPTED-PATCH MEAN	REPEAT VARIATION	REPEAT CORRELATION	VALIDATION AUC
80%	±2 pts	0.96	0.75–0.86

Exhibit 3 **Task-specific, route-blinded quality measurement.**

Framework validation on ground-truth software tasks with executable outcomes · Source: Faros evaluation framework

among patches receiving the same binary failure outcome, rubric scores recovered the model capability ordering at +0.94 Spearman correlation.

Executable tests provide complementary behavioral evidence when a reproducible environment and relevant coverage are available. The rubric measures quality that a pass/fail test outcome cannot distinguish; tests catch what only running the code shows. The Agentic Rubric provides a consistent, graded quality measure across the Route Index cohort.

FINDING 1 · OPEN MODELS

Open-model routes recorded the highest mean quality in both releases.

In each Route Index release, the highest mean-quality results came from routes running open models.[1,3]

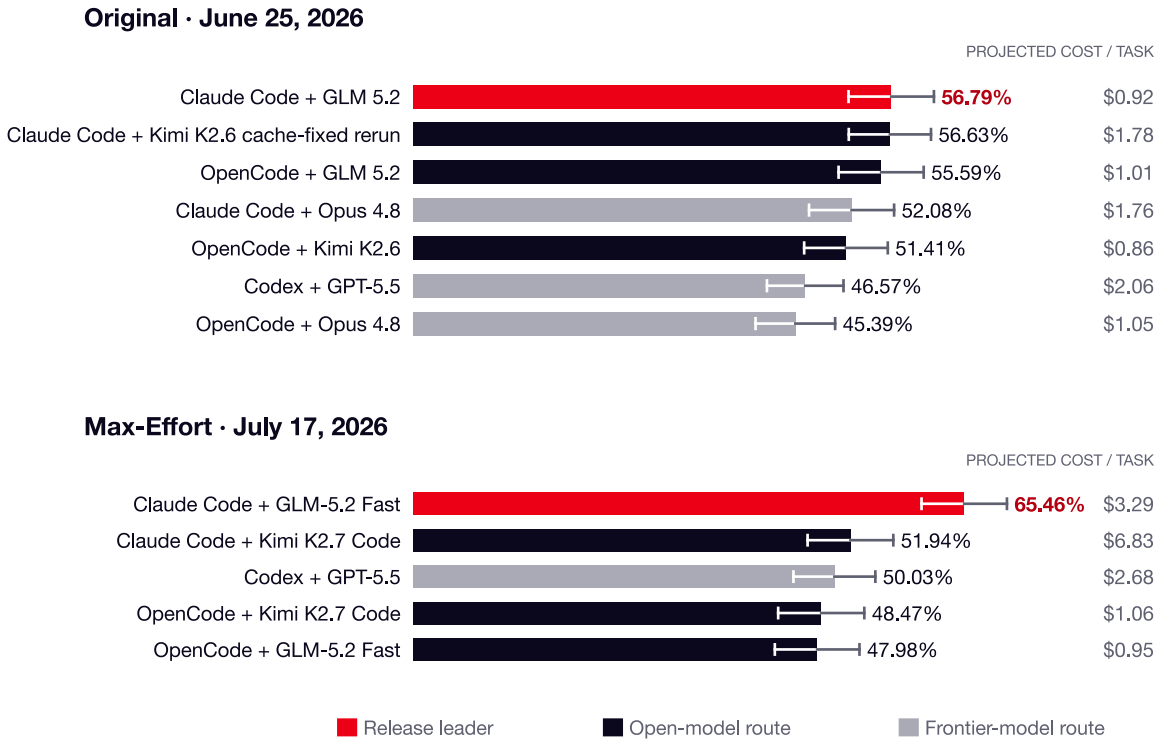

Exhibit 4 **Open-model routes led both releases.**

Mean Agentic Rubric score with marginal 95% interval whiskers and mean projected cost per task · 211 tasks per route · Paired route-to-route comparisons carry their own intervals · Source: Faros Route Index

Claude Code + GLM 5.2 recorded a 56.79% mean rubric score at a projected cost of $0.92 per task, with the Claude Code + Kimi K2.6 cache-fixed rerun[1] just 0.16 points behind at 56.63%. The two routes are effectively tied at the portfolio level; the paired 95% interval on the task-level difference runs from −4.6 to +5.0 points, and Finding 3 shows what that near-tie contains at task level. The top three positions by mean score were all open-model routes. The strongest frontier-model route, Claude Code + Opus 4.8, trailed the leader by 4.71 points.

Claude Code + GLM-5.2 Fast recorded a 65.46% mean rubric score in the Max-Effort release, 13.52 points above the next route, with a paired 95% interval on that lead of 8.0 to 19.1 points.[3] It also recorded the highest task-winner share in the release at 34.43%, leading the runner-up on 97 tasks, tying on 65, and trailing on 49. The strongest frontier-model route, Codex + GPT-5.5, finished third, 15.43 points behind the leader.

In these dated releases, open-model routes led on mean rubric score: licensing category alone is not a reliable proxy for observed route performance. An open model operating inside the right route produced a leading quality and economic position on real enterprise engineering work.

FINDING 2 · SAME-MODEL ROUTE BUNDLES

The same model label produced different results in different routes.

Five comparisons keep the model label constant while changing the surrounding route bundle. The observed differences show why the model name alone is not a deployment identity.

Route-bundle differences are computed before display rounding, and each pairing's paired 95% interval is computed on the task-level differences across the 211 shared tasks. Across these five pairings, the same model label produced route-bundle differences ranging from 1.20 to 17.48 quality points. Paired task-level intervals identify clear route-bundle separation in three comparisons: Kimi K2.6, Opus 4.8, and GLM-5.2 Fast. For the Original pairs, that separation holds under the familywise adjustment reported in Methods. The other two pairings produced smaller observed differences whose intervals include zero. The size of the observed bundle difference varied across model labels. That variation is why the bundle effect has to be measured rather than assumed.

These are complete route-bundle comparisons. They do not assign the difference to one isolated component. They show that the assembled route, including the harness and declared operating configuration, is the meaningful system to measure.

[1]The corrected rerun that replaced the original Kimi K2.6 attempt after a provider caching issue; see the source study.[1]

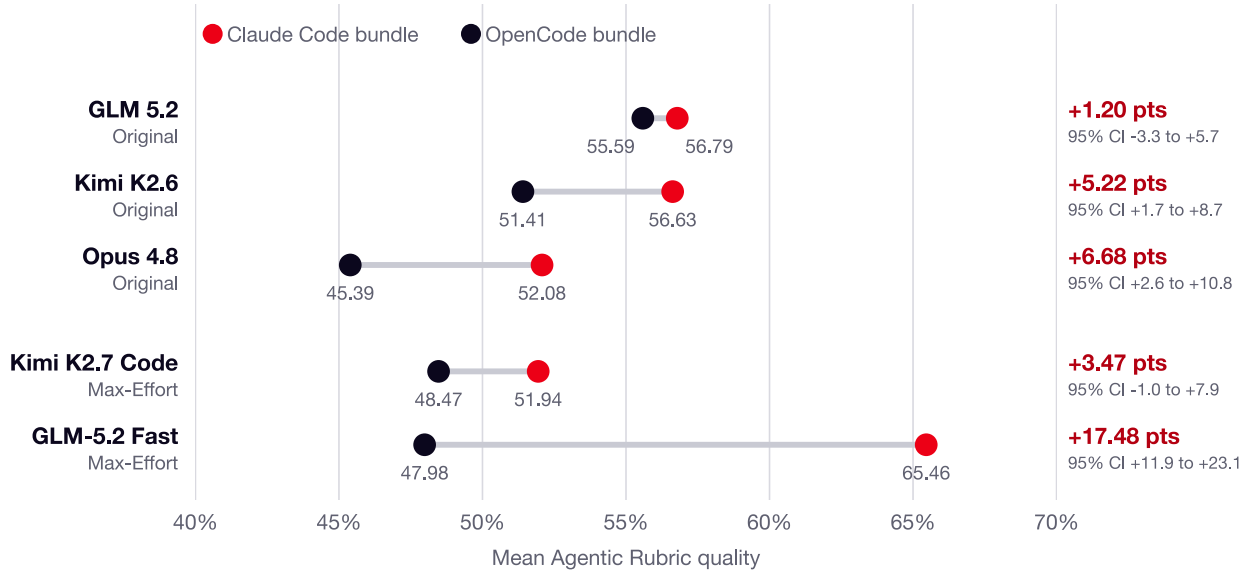

Exhibit 5 **The same model is not the same system.**

Five same-model route pairs with paired 95% intervals · 211 paired tasks per comparison · Quality = execution-free Agentic Rubric score · Source: Faros Route Index

Buying access to a model does not specify how that model will investigate a repository, manage its work, call tools, recover from errors, or turn a plan into a patch. Those behaviors come from the assembled route, and the assembled route is what Time Machine measures.

FINDING 3 · AVERAGES AND TASK OUTCOMES

Similar portfolio averages contained different task-level outcomes.

A route average answers one useful question: how did this configuration perform across the whole cohort? The paired task matrix answers another: on which tasks did each route produce the stronger result? The Original release's top two routes illustrate the distinction: their portfolio means were separated by 0.16 points, a portfolio-level tie (paired 95% interval −4.6 to +5.0 points), yet the 211 paired tasks split three ways.

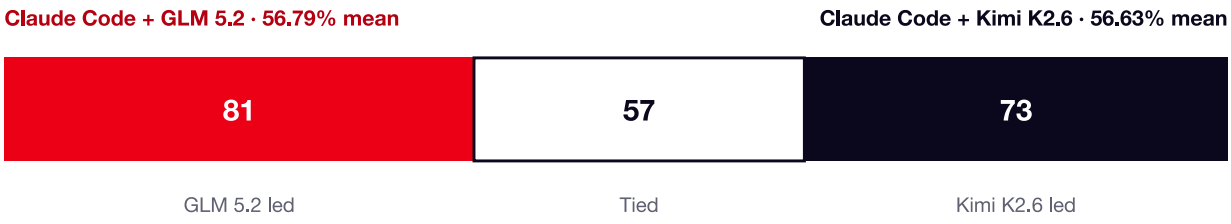

Exhibit 6 **Similar averages, different task outcomes.**

The 0.16-point portfolio near-tie contained 81 GLM-leading tasks, 57 ties, and 73 Kimi-leading tasks · 211 paired tasks · Source: Faros Route Index

The average establishes the portfolio result. The task matrix preserves the distribution of that result across the workload. For organizations, both views matter: the aggregate informs a broad route decision, while the paired evidence shows where the route outcomes differed.

Time Machine starts with representative historical work because the decision depends on how candidate routes respond to the work an organization needs them to perform, not only on a cohort-wide rank.

FINDING 4 · QUALITY AND COST

Quality and projected cost form one decision surface.

AI coding routes consume different amounts of provider capacity and compute. The useful operating question is which routes create the strongest quality-cost positions for the work.

In the Original release, Claude Code + GLM 5.2 combined the highest mean quality, 56.79%, with a projected cost of $0.92 per task, while OpenCode + Kimi K2.6 provided a lower-cost position at $0.86 with 51.41% mean quality. Codex + GPT-5.5 recorded 46.57% mean quality at $2.06 per task, lower quality at more than twice the leader's projected cost.

In the Max-Effort release, the observed quality-cost frontier extended across four operating points, from OpenCode + GLM-5.2 Fast at 47.98% and $0.95 to Claude Code + GLM-5.2 Fast at 65.46% and $3.29. The quality leader cost less per task than the second-ranked route, which recorded 51.94% mean quality at $6.83.

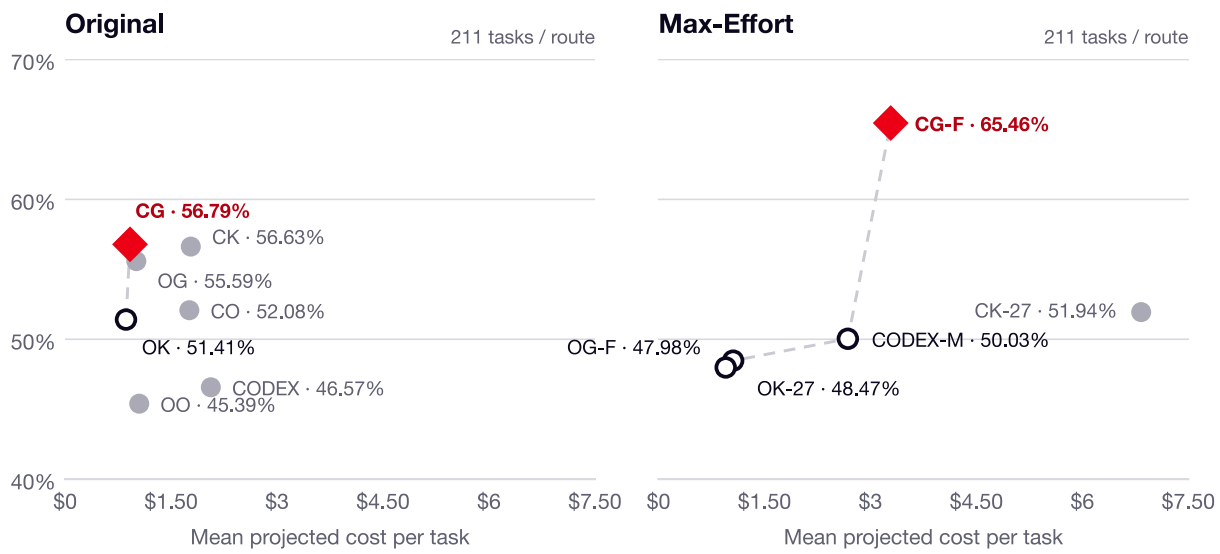

Exhibit 7 **Quality and projected cost are one decision surface.**

Mean quality and mean projected cost for all 12 dated route configurations · Identical axes per panel · Dashed line connects each release's observed Pareto frontier · Red diamond = release quality leader, hollow circle = frontier route, gray = dominated route · Codes: CG/OG = Claude Code/OpenCode + GLM 5.2 · CK/OK = Claude Code/OpenCode + Kimi K2.6 · CO/OO = Claude Code/OpenCode + Opus 4.8 · CODEX, CODEX-M = Codex + GPT-5.5 · CG-F/OG-F = Claude Code/OpenCode + GLM-5.2 Fast · CK-27/OK-27 = Claude Code/OpenCode + Kimi K2.7 Code · Source: Faros Route Index

This joint view turns route selection into an engineering operating decision. Organizations can compare quality and economics together and choose an operating point aligned with the work and the required outcome.

SECTION 5 · WHAT THE EVIDENCE CHANGES

Five practical conclusions.

Evaluate the complete route. Version the model, harness, provider, runner configuration, context, tools, and operating controls that produced the result. A model name alone is not enough to reproduce or govern a deployment decision.

Use representative local work. General benchmarks identify promising candidates. Historical tasks from the organization establish how those candidates perform on its repositories, conventions, architecture, and work mix.

Compare alternatives on the same tasks. Paired historical replay gives every candidate the same engineering problems and relevant repository states. This produces direct route-to-route evidence rather than comparisons confounded by different task samples.

Preserve both portfolio and task-level evidence. Route averages support broad operating decisions. The route-task matrix shows how those averages were assembled and provides the evidence needed to examine meaningful engineering scopes.

Join quality and economics. Quality and projected cost belong in the same decision. The target is the operating point that satisfies the organization's engineering and economic objective.

These principles shift AI coding evaluation from a one-time procurement exercise to an evidence-maintenance discipline.

SECTION 6 · FROM EVIDENCE TO GOVERNED ACTION

The control model.

Time Machine becomes operationally valuable when historical comparison informs how an organization manages AI coding systems. The Faros control model connects that evidence to two outcomes: recommendations and human-approved policy proposals.

Control	Function	Evidence path	Decision
Recommendation	Advisory diagnosis and suggested action	Relevant usage, session, and engineering signals are re-analyzed	A person decides whether and how to act
Policy	Enforceable configuration proposal	A concrete alternative is replayed and scored through Time Machine	A person approves before the policy takes effect

Both begin with the organization's own usage evidence and a scoped diagnosis. The diagnosis identifies a candidate opportunity. When the opportunity has a concrete configuration lever, such as the route assigned to a repository or work type, Time Machine can compare alternatives on historical ground-truth tasks. Quality, projected cost, and sample evidence support the resulting policy proposal. When the opportunity calls for a human practice rather than a configuration change, the relevant evidence is re-analyzed and surfaced as a recommendation. Examples include improving task scope, context management, tool leverage, or review calibration. In both paths, Time Machine supplies the historical simulation and measurement layer, and the person remains the decision-maker.

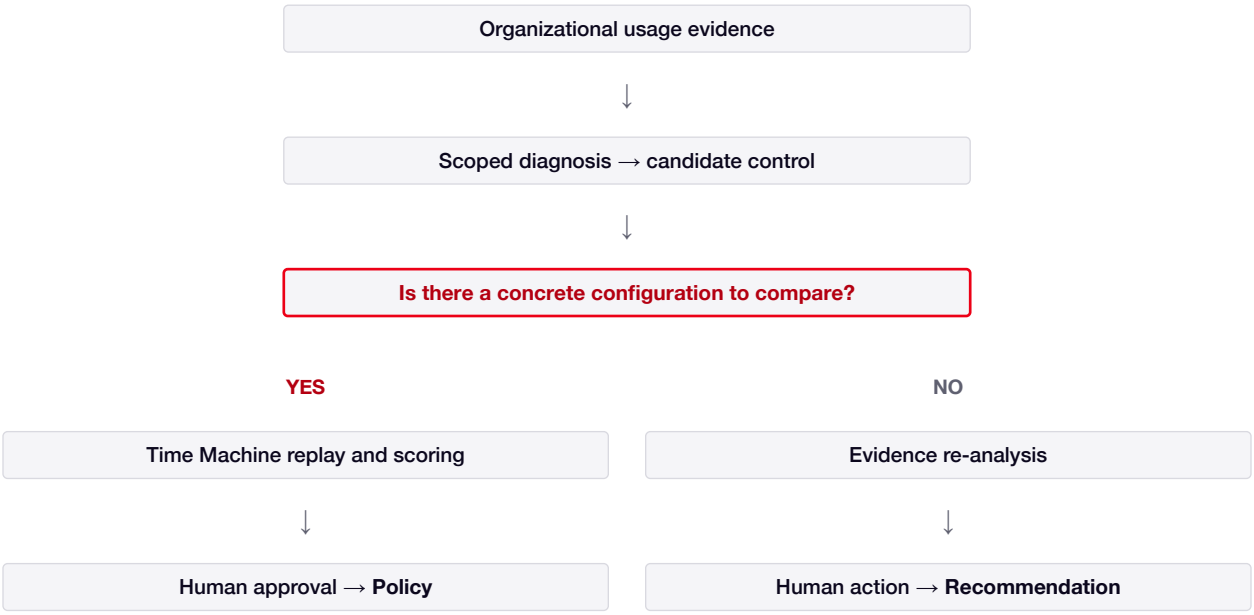

Exhibit 8 **From observation to governed action.**

Time Machine validates concrete alternatives; a person approves every action · Source: Faros control model

SECTION 7 · THE OPERATING DISCIPLINE

A repeatable evaluation cycle, not a permanent winner.

Models, harnesses, provider paths, costs, and organizational workloads continue to change. A durable operating model therefore depends on a repeatable evaluation cycle rather than a permanent winner.

Define the deployed unit. Record the complete route and effective context as one versioned decision identity. **Select representative historical work.** Use completed tasks that reflect the repositories, task descriptions, architecture, conventions, and work mix the system will encounter. **Replay declared alternatives** on the same tasks under fixed controls. **Measure quality and economics**, preserving the task-specific quality result and economic evidence for every candidate attempt. **Decide at the appropriate scope**, using portfolio results for broad route choices and task-level evidence for repository, work-type, or team decisions. **Govern the action**, turning validated configuration choices into human-approved policies and evidence-grounded practice changes into recommendations. **Refresh the evidence** when a meaningful part of the route, workload, controls, or economics changes.

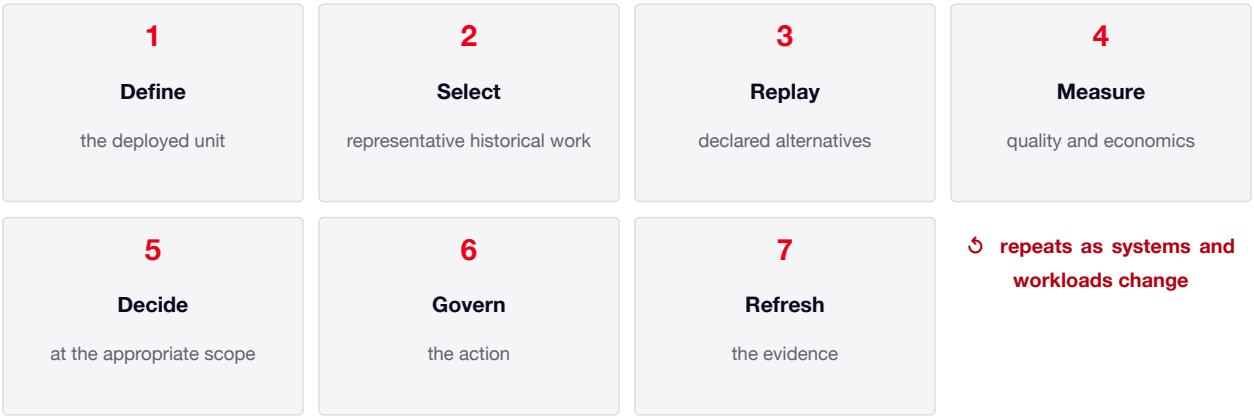

Exhibit 9 **The route-evaluation operating cycle.**

Route evaluation is a maintained engineering capability, not a one-time leaderboard choice · Source: Faros Research operating framework

Applying the framework to an organization's own work

The Route Index demonstrates Time Machine on Faros engineering work. The same evaluation discipline can be applied to the repositories and tasks that define another organization. An organization-specific comparison begins with the decisions the organization needs to make:

- Which complete routes perform best on our code?
- Which routes create the strongest quality-cost position for each work type?
- Where does a lower-cost route preserve the quality we require?
- Which repository or task scopes respond differently to the same route?
- Which declared context changes improve a fixed route?
- Which configuration decisions have sufficient evidence to become governed policies?

Completed engineering work supplies the evaluation corpus, historical repository states the environment, and accepted changes the task-specific evidence. The organization's priorities determine the decision scope. Candidate routes are selected using evidence from the work the organization expects them to perform, not because they are generally popular or lead an unrelated benchmark.

For engineering leaders, the result is a maintained record of what was compared, which work represented the decision, how each route performed, what the quality and economics showed, which control was proposed, and when the evidence should be refreshed. That record turns AI coding adoption into an accountable engineering system.

CONCLUSION

Engineering outcomes are produced by complete systems.

The AI coding market is organized around model releases. Engineering outcomes are produced by complete systems. The Faros Route Index makes that distinction visible. Across 211 real historical engineering tasks and 2,532 route-task results:

- open-model routes recorded the highest mean rubric scores in both releases, with the strongest frontier-model route 4.71 and 15.43 points behind the respective leaders;
- same-model route bundles differed by as little as 1.20 and as much as 17.48 quality points, and paired intervals showed clear route-bundle separation in three of the five pairings;
- task-level outcomes added information beyond portfolio averages; and
- the quality leader in each release did not carry the highest projected cost.

Faros Time Machine made these comparisons possible by converting completed engineering work into controlled historical evidence. It preserves the task, restores the relevant repository state, withholds the accepted solution, executes declared alternatives, measures task-specific quality, and retains the economic evidence required for a decision.

The resulting capability connects three activities that organizations have previously treated separately: observing how AI coding systems are used; comparing what alternative systems produced on the same historical work; and governing the configuration or practice that follows.

The model is not the system. The complete route, operating in context, is the unit that reaches the repository, and representative historical work is the evidence required to evaluate it.

Measure the route, and keep the evidence current.

METHODS AND DEFINITIONS

Study corpus. A fixed cohort of 211 anonymized historical Faros engineering tasks represented in the Faros Route Index, spanning 12 Faros repositories. Each task derives from completed engineering work and is paired with the relevant historical repository state. The cohort was built from a selected pool of 500 historical Faros work instances across the 12 repositories.[1] In the Original release, a task entered the strict headline cohort only if every one of the seven routes produced a valid implementation and a judge score for it; 211 tasks met that requirement, and both releases publish a complete route-task matrix on that fixed cohort.[1,3] Invalid, no-diff, failure, and timeout outcomes are tracked separately in run telemetry and are not mixed into the headline means. Gap-fix and recovery reruns were used where needed to repair telemetry or completion gaps, and the Kimi K2.6 route was rerun to correct a provider

caching issue; the earlier affected run was excluded from the published release.[1] The selection flow is: 500 selected instances → route execution → validity and scoring gates → 211-task strict cohort → published results. By work type, the cohort contains 61 maintenance, 54 feature, 49 infrastructure and developer-experience, 27 bug-fix, and 20 other tasks.[1]

Releases. The Original release (`faros-engineering-2026-06-25`, June 25, 2026) compares seven declared routes: 1,477 route-task results. The Max-Effort release (`faros-engineering-max-2026-07-17`, July 17, 2026) compares five: 1,055 results. Combined: 2,532 results across 12 dated route configurations; the route label Codex + GPT-5.5 appears in both dated releases, and each dated entry is counted separately. The Route Index records the Original release under Replay Protocol v1 and the Max-Effort release under v2. The published evaluation definitions are unchanged between the two; v2 explicitly records that route versions and configurations are frozen separately for each dated official run. The findings in this report are within-release comparisons: the releases are dated route configurations, so this report makes no like-for-like quality comparison across releases.

Route. The complete declared model, harness, provider, and runner configuration used for a comparison. Context and the task cohort are held fixed within a route-comparison release.

Route configurations. The routes compared in this report, with the served model, provider, and effort setting recorded in the Route Index catalog.[3] Remaining runner-configuration detail is recorded with each dated release in the catalog.

Release	Route	Harness	Served model	Provider	Effort
Original	Claude Code + GLM 5.2	Claude Code	GLM 5.2	Fireworks	Default
Original	Claude Code + Kimi K2.6 cache-fixed rerun	Claude Code	Kimi K2.6 Turbo	Fireworks	Default
Original	OpenCode + GLM 5.2	OpenCode	GLM 5.2	Fireworks	Default
Original	Claude Code + Opus 4.8	Claude Code	Opus 4.8	Bedrock	High
Original	OpenCode + Kimi K2.6	OpenCode	Kimi K2.6 Turbo	Fireworks	Default
Original	Codex + GPT-5.5	Codex	GPT-5.5	OpenAI	High
Original	OpenCode + Opus 4.8	OpenCode	Opus 4.8	Bedrock	High
Max-Effort	Claude Code + GLM-5.2 Fast	Claude Code	GLM-5.2 Fast	Fireworks	Maximum
Max-Effort	Claude Code + Kimi K2.7 Code	Claude Code	Kimi K2.7 Code	Fireworks	Maximum
Max-Effort	Codex + GPT-5.5	Codex	GPT-5.5	OpenAI	Maximum
Max-Effort	OpenCode + Kimi K2.7 Code	OpenCode	Kimi K2.7 Code	Fireworks	Maximum
Max-Effort	OpenCode + GLM-5.2 Fast	OpenCode	GLM-5.2 Fast	Fireworks	Maximum

Route-task result. One candidate route's attempted resolution of one historical task, with its associated quality and projected-cost evidence. Each route-task cell contains one published implementation and one judge score; results are not averages over repeated attempts.[1]

Agentic Rubric score. The weighted percentage of task-specific Agentic Rubric criteria satisfied by a candidate change, reported on a 0–100% scale. The judge is blind to route identity. The reported score is execution-free; executable tests provide complementary evidence when they are available.

Projected cost. An estimate of the route-level model expenditure associated with a task result using the applicable usage and rate evidence for the release.

Aggregation and uncertainty. Mean rubric scores and mean projected cost are computed across the 211 paired tasks for each route. Task-winner shares split ties fractionally. Direct task-level comparisons preserve the paired cohort. Each route mean carries a marginal 95% confidence interval, shown in Exhibit 4. Route-to-route statements use paired task-level differences with their own 95% intervals; a difference is described as clearly separated only when its paired interval excludes zero. For the Original release's three same-model comparisons, a sensitivity analysis adjusting jointly for the three contrasts gives simultaneous intervals of -4.3 to $+6.7$ points for GLM 5.2, $+0.9$ to $+9.5$ for Kimi K2.6, and $+1.7$ to $+11.7$ for Opus 4.8; the Kimi K2.6 and Opus 4.8 separations remain clear after adjustment.

Scope of the evidence. All 211 tasks derive from one organization's repositories; the results characterize route performance on Faros engineering work, and other organizations should expect different orderings on their own work, which is why the operating framework starts from an organization's own history. The headline cohort is conditioned on completion: it contains the tasks for which every compared route produced a valid implementation and score, while excluded and failed outcomes are tracked separately and do not enter the headline means. The primary quality measure is execution-free: it measures rubric-weighted alignment with task-specific criteria, and executable tests add behavioral evidence where they are available. Each route-task cell contains a single published result; repeated-judgment variation for the scoring framework is documented at approximately ± 2 points.[2] Route comparisons are bundle-level: they attribute differences to complete declared configurations, not to any single component. The Route Index is an operated benchmark: each release is a dated, versioned measurement run of the declared routes on the fixed cohort. The evidence window closes at the July 17, 2026 Max-Effort release; later Route Index releases are new dated evidence.

Data protection. The report publishes anonymized task-level aggregates and route-level results. Repository identities, work-item text, candidate patches, accepted changes, prompts, and customer data are not included.

Terminology. *Historical task*: a completed engineering work item paired with its relevant pre-change repository state. *Treatment*: the declared route or context variable changed in a controlled comparison. *Route comparison*: the complete route changes while the task cohort, context, evaluation, and run controls remain fixed. *Context comparison*: the effective context changes while the route, task cohort, evaluation, and run controls remain fixed. *Control*: an evidence-backed recommendation or human-approved policy.

REFERENCES

1. Faros Research. Open Models vs. Frontier Models: AI Coding Routes on Real Engineering Work.
2. Faros Research. AI Model Routing Evaluation Framework.
3. Faros Research. Faros Route Index.
4. Jimenez, C. E., et al. SWE-bench: Can Language Models Resolve Real-world GitHub Issues?
5. National Institute of Standards and Technology. NIST AI 800-2: Practices for Automated Benchmark Evaluations of Language Models, Initial Public Draft.

About Faros Research

Faros Research publishes evidence and methods for measuring AI-assisted software engineering. This report was produced by the Faros Research team. Explore the current Route Index release at Faros Route Index, and learn more at faros.ai.

Document information

Version 1.0 · July 2026. This report covers Faros Route Index evidence published through July 17, 2026. Faros Research will date material corrections, preserve prior public versions, and identify changes on the public report page. Updated models, routes, cohorts, or evaluation methods constitute new evidence, not silent revisions to this analysis.