THE ECONOMICS OF EXPANDING PRODUCTIVE CAPABILITY

AGI ALPHA FLYWHEEL

From Useful Work to Reusable Productive Capability

SUCCESSOR Ω Harness ACTIVATE. EVOLVE. SCALE.

Powered by TRANSFORMATIONAL INTELLIGENCE ∞

A preregistered local capability-construction study
and market-data-grounded comparison with traditional human work

Prepared with AI assistance for repository and scientific review.
Evidence edition: 25 October 2026. No claim of peer review or production admission.

Abstract

An employer needs a release plan, a checked report or a source-backed brief. The commercial question is what accepted result arrives, when, at what total cost, with how much attention—and what useful method remains for the next job. We study that transaction using the existing AGI ALPHA workbench and strategy worker, a separately authored evidence adapter, frozen acceptance contracts and official labor-market data. All 42 primary jobs across seven bounded digital-work families were accepted by separately implemented automated checks. Median local turnaround ranged from 0.099 to 1.301 seconds by family. A connected planning journey used actual parent artifacts to construct lookahead and repair methods; inheritance avoided reconstruction work. Yet the retained repair candidate lost to the available specialist on all 12 fresh transfer instances. Incremental source repair used 1.036 times the CPU of full recomputation, and no discovery cycle admitted an improvement. Three batches at each of concurrency 1, 2 and 4 accepted 108/108 outputs, with 3.176 times the median finite-batch throughput at four processes. These small local computations are not a production fleet. Official wages and employer costs support transparent break-even and sensitivity models; matched human durations, complete Agent cost and employer attention remain unmeasured. Therefore a human-relative monetary advantage, cash reinvestment and self-acceleration are not established. The contribution is an inspectable, bounded path from useful delivery to retained executable methods, fresh transfer, strong-tool fallback and economic admission that preserves negative findings.

Primary benchmark: competent traditional human work with ordinary non-AI tools.
Mechanism experiments are separate. No participant recruitment or financial execution occurred.

1 The employer's decision: deliver this work, then retain what helps

Consider an operations manager preparing a small release. Six jobs have dependencies, durations, crew requirements and a deadline. The manager wants an editable schedule and evidence that it respects those constraints. A requirement then changes; later, a crew outage invalidates part of the plan. The manager needs a revised schedule that preserves already committed work. The next release has different inputs. The question is whether the system can deliver all of this reliably, and whether anything learned or constructed makes the next delivery better after its full cost.

Our connected demonstration produces those artifacts. It first delivers a plan finishing at slot 15, then a plan finishing at slot 16 after a duration change, then a valid repair finishing at slot 21 after a resource-calendar interruption. These slots are declared scenario inputs, not measured human hours. The repair preserves the two locked job starts. An independent implementation checks the schedule against the actual employer brief. The completed files include readable HTML, editable CSV, the input requests, source identities, worker outputs, acceptance decisions and accounting records. A dashboard is only an interface to this work.

Table 1. The same employer transaction, with different legitimate internal methods. The human column describes the comparator workflow; it is not a newly observed trial.

Employer requirement	Traditional human workflow	AGI ALPHA observed local workflow
Create a valid plan	Interpret the brief; use project software, a spreadsheet, professional knowledge and reusable templates.	Validate a typed request; run the existing strategy portfolio and available native solver; export schedule and checks.
Respond to change	Compare versions, inspect dependencies, preserve commitments, revise and review.	Read the actual parent artifact; identify changed constraints; construct and test repair; keep the best qualified available result.
Accept the delivery	Check correctness, completeness and usability against the same brief; correct defects.	A separately authored checker verifies constraints, source binding, preserved work and output structure.
Obtain later value	Reuse experience, templates and procedures where applicable.	Retain executable strategy artifacts, exact evidence dependencies and regression checks; invoke them on fresh tasks and measure their contribution.
Pay the full cost	Labor, review, correction and attributable employer costs.	Execution, review, retries, construction, setup, support, maintenance and remaining human attention; missing components remain unknown.

We define *computer-mediated work* as work performed through digital interfaces using a keyboard, mouse or equivalent input, with results delivered or actions completed through software. This definition includes a substantial range of useful activity without implying that every responsibility in a salaried role is automatable. Accountable decisions, stakeholder negotiation, ambiguous requirements, confidential access, physical execution and professional judgment remain outside these experiments.

The *AGI ALPHA FLYWHEEL advantage* is the complete system's measured advantage over the stated traditional-human comparator at accepted quality. It is an operational definition, not a requirement that an advantage exist. Retaining code can be useful without proving faster work; faster computation can occur without lower employer cost; lower cost can occur without

cash savings; and one improved method does not establish accelerating improvement. The paper reports these as different outcomes.

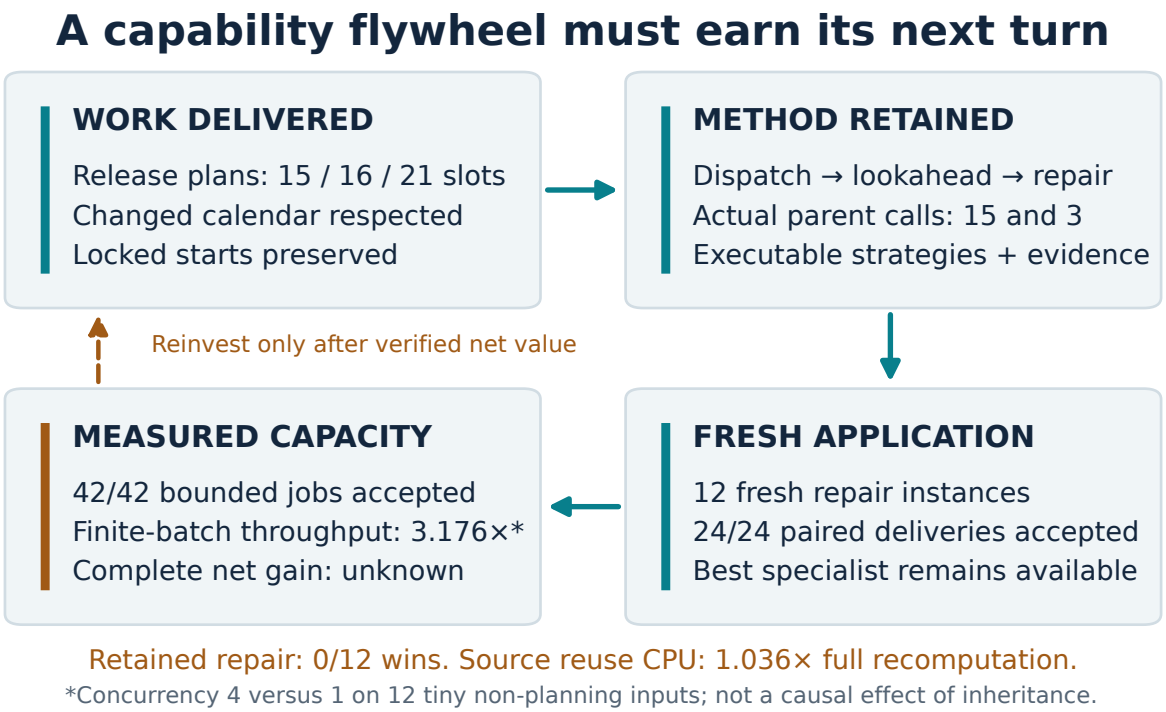


Figure 1. The signature demonstration links delivered work, retained executable methods, fresh application and measured capacity. Each arrow requires evidence. The capacity measurement is a separate load test, not a causal consequence of inheritance. The dashed reinvestment step remains conditional on net economic value. Negative transfer and reuse results are part of the figure.

2 Study identity, preregistration and the measured system

This is a new experiment, separate from the completed *Economics of Autonomous Work* edition associated with pull request #4901. That edition's 69 frozen inputs, results, limitations and canonical PDF remain byte-identical. We neither broaden its task claims retrospectively nor reinterpret its earlier software comparisons as observations of human work. Its publication track continues independently. The new study is coordinated through issue #4850.

The new protocol froze 110 implementation, generator, evaluator, source and economic-input files at 12:31:23 UTC on 25 October 2026. A durable checkpoint and public issue notice preceded confirmation. The freeze SHA-256 is

`55fe389e077028c1439cf3dbdadc09164455df00865db471ddf03bd1eb99e282.`

All 110 files were independently checked before and after confirmation. Fresh seeds, instance counts, acceptance rules, cost boundaries and analysis were declared in advance. Development attempts, including sandbox failures and an uninformative planning topology corrected before freeze, are retained. No failed confirmation case was replaced and no confirmation retry was added.

2.1 Complete available route, with an explicit execution boundary

The implementation uses AGI ALPHA main source `f836f06e8351e64db1da770e16e304b5feb01f28`. The maintained workbench supplies bounded analysis, testing, documentation and numerical functions. The original strategy worker supplies planning, construction accounting and the available SciPy/HiGHS specialist. A new, reviewed study adapter supplies exact source-backed briefs and printable HTML presentations through the workbench's actual extension interface. Selection of a strong existing method is part of the product's value; no specialist is removed to manufacture a learning gain.[21, 22]

This configuration is the available *local productive route* through the Agent's source. It is not an authenticated installed enterprise or a general computer-use trial. Task-time execution is deterministic and makes zero model/API calls. The research implementation was authored with a capable host AI assistant; its exact provider snapshot, total token bill and full authoring effort are not exposed to the experiment. Those costs are unknown, not zero. The supported model identity is the session label GPT-6 Astra Pro, not a reproducible provider snapshot. No later provider upgrade is presumed equivalent.

The native Journal path would require an already approved operator home and authority. None was supplied for this continuation, and the update authorized no new credentials or security changes. The authenticated browser-to-executor route is therefore **NOT RUN**. Pure local invocations exercise real maintained code without creating such authority. The browser companion clearly separates that local computation from recorded planning, modeled economics and the unexecuted approved-executor route.

2.2 Acceptance, comparison and statistical scope

Each primary request has a useful output, a fixed acceptance contract and a one-workday employer deadline; smaller system limits bound local execution. Correctness, completeness, usability, provenance and material-error conditions are frozen before execution. Exact structured jobs require zero material contract violations. Acceptance checks are separately implemented in Python; they are outside the method search being assessed. The evaluators are independent implementations within the same study team, not blinded human customers or an independent institution. Document usability is checked structurally, and the publication artifacts are visually inspected; subjective professional quality is not thereby established.

The primary portfolio contains 42 jobs: six instances in each of seven bounded families, with two instances at each of three complexity strata. The portfolio uses eight principal templates. It does not contain 42 independent occupations. Fresh inputs vary content, size, dependencies or exceptions. Randomized order is used within the non-planning and planning tracks. We do not claim a single global random ordering. Human workers in the economic comparator are allowed efficient ordinary software, templates and experience; there is no assumption of repeated novice work. Human AI use is unknown in most wage sources and explicitly recorded when published performance evidence does not establish a strict non-AI control.

Descriptive Wilson intervals and task-level bootstrap intervals quantify the small observed samples. They do not correct for purposive task selection, shared templates, synthetic inputs, source knowledge or common evaluation assumptions. Load repetitions, review mutations and mechanism attempts are not added to the primary-job denominator.

3 What work was delivered?

All 42 primary jobs were accepted on their first operating attempt. The descriptive 95% Wilson interval is 91.62–100%; the sample design does not license a population-wide reliability estimate.

Figure 2 shows every observed turnaround. Table 2 names the actual deliverables and what they exclude. The source passages are from pinned public repository documents; numeric inputs, software instances and project scenarios are synthetic but fully shareable. No private employer record or paid customer transaction is represented as observed.

Table 2. Bounded primary portfolio. Each row has six accepted fresh instances. Turnaround medians include the local delivery boundary described in the text.

Family	Actual finished work and boundary	Median (s)
Software/testing	Three closed-interval defect repairs with runnable checks; three exact sorted-integer deduplication implementations/comparisons. No arbitrary code repair.	0.103
Structured analysis	Exact two-column integer CSV aggregation, editable summary and HTML report. No open-ended business inference.	0.124
Research/synthesis	A traceable brief containing exactly the requested public source passages, with source links and hashes. No autonomous literature search or semantic truth adjudication.	0.113
Documents and slides	A source-backed brief and printable HTML slide document. No professional design rating or general PowerPoint editing.	0.123
Planning/operations	Six-, nine- or twelve-job schedules respecting declared dependencies, resources and deadlines. No observed physical completion.	1.301
Scientific computation	Exact polynomial integral and honest quadrature/error reporting within the supported grammar. No scientific discovery claim.	0.099
Web/documentation	Executable URI/UTF-8 documentation examples, corrected text and per-example checks. No general website audit.	0.104

The median over the 36 non-planning deliveries is approximately 0.112 seconds; the planning median is 1.301 seconds. These are deliberately kept separate. Non-planning timing includes producer launch, execution, independent acceptance and export through receipt creation, excluding the final receipt-serialization tail. Planning records the entire child-process launch-through-exit interval, including external checking and exported schedules. CPU includes waited descendants; threaded native-solver CPU may exceed wall time. Full source construction, session setup, employer review, deployment and maintenance are outside these task clocks and require separate accounting.

A numerical report is acceptable when it computes the exact reference and truthfully reports the approximation's error and tolerance decision. Acceptance therefore does not mean that every numerical approximation met its requested tolerance. Similarly, an exception report is valuable because it exposes a problem accurately, not because it declares the underlying business issue solved. These distinctions make the acceptance contract useful to an employer.

For each job, the companion retains the request, input, output artifacts, independent verdict and complete local receipt. Unknown dollar cost and human attention appear as null fields. A successful run is not silently converted into a free run. No human-time ratio or dollar saving is estimated directly from the milliseconds in Figure 2.

4 A plan, a change, and an executable repair method

Consider an employer preparing a public evidence release. Several preparatory jobs compete for a small team; integration must precede review and packaging. The employer needs a schedule that respects those dependencies and the available capacity. Then integration takes longer than expected, and a future service window becomes unavailable. The useful result is a repaired schedule that retains valid prior decisions and explains what changed. A lineage diagram alone would not perform this work.

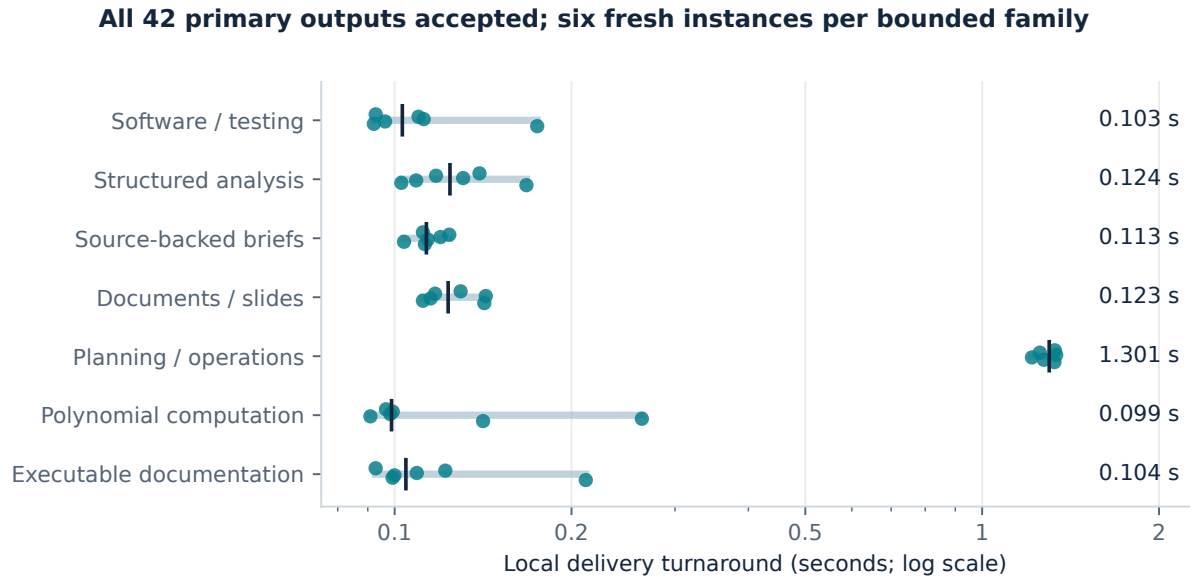
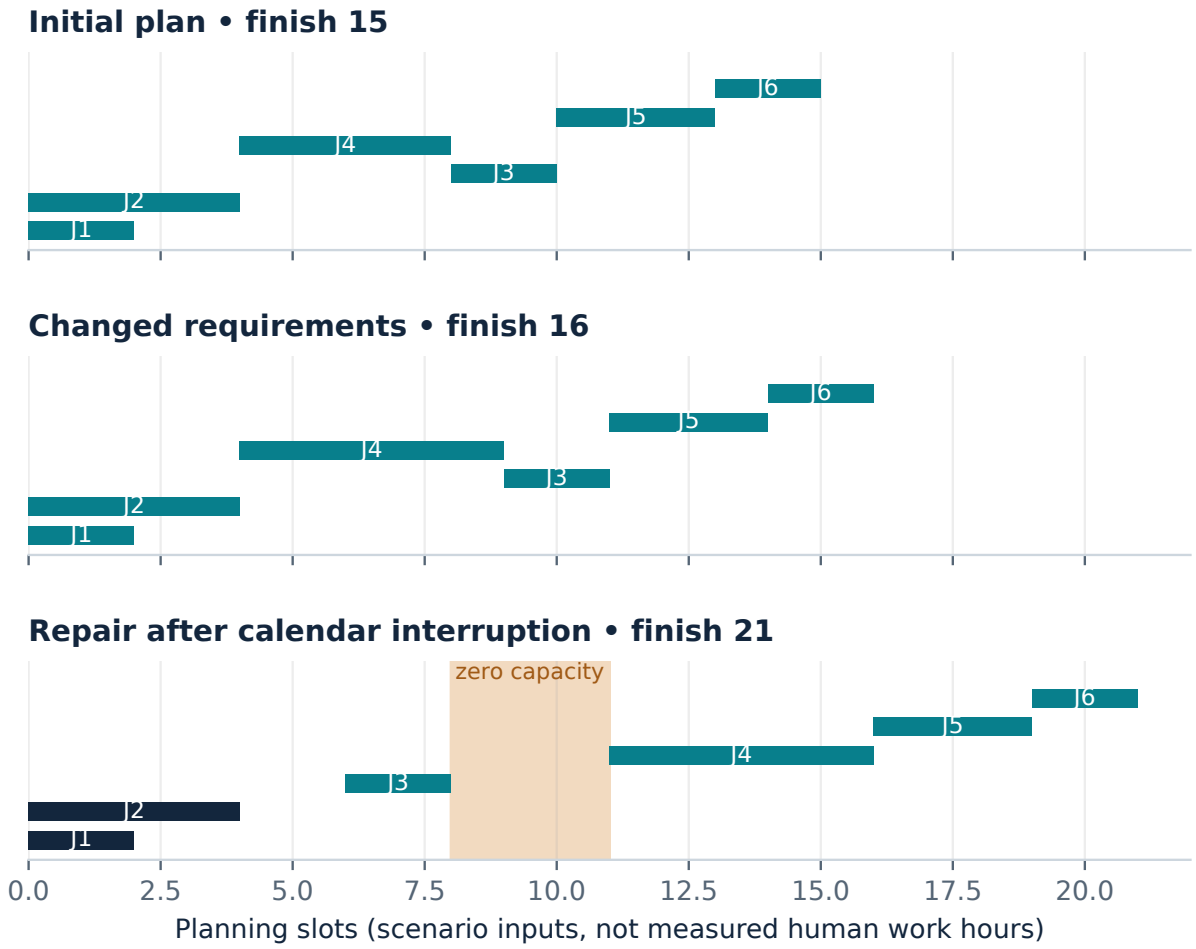


Figure 2. Observed local delivery distributions. Six dots per bounded family, with a median marker. All primary instances are shown. Fast local file production is not evidence of equally fast open-ended professional work, nor a human speed ratio.

We executed this sequence through the existing, source-pinned Strategy Succession worker. Its fixed interface accepts a bounded project, invokes the ordinary scheduling portfolio, offers the existing local SciPy/HiGHS specialists, checks returned schedules, and produces a delivery pack. The study added independent acceptance and measurement around that interface; it did not replace the worker's scientific source, disable strong specialists, or create an operator identity. The fixed strategy grammar contains dispatch rules, bounded lookahead and repair wrappers. A retained capability is therefore an executable strategy abstract syntax tree interpreted by the existing worker, with its source binding, applicability, qualification, costs and rollback status retained.

What the acceptance contract covers. The independent Python oracle imports neither the producer nor its optimizer. It reconstructs the dependency order and resource usage from the submitted brief, checks every job exactly once, verifies durations, releases, calendar capacity, locked starts, deadline and movement limit, and independently derives the reported lower bounds. It checks makespan and total start-time movement directly from the schedule. A solver success flag is insufficient. Equality to a valid lower bound supports the corresponding makespan claim; a feasible schedule that exceeds the bound is not presented as proven optimal. All current proof and authority fields must remain empty. This is independently implemented engineering acceptance within a common research environment, not an externally administered human trial.

The six primary planning jobs span six, nine and twelve jobs, with two fresh instances at each size and randomized execution order. All six were accepted. Median complete launch-to-exit turnaround was 1.300576 seconds, with a range of 1.215540–1.336691 seconds. Median worker and waited-descendant CPU was 4.002799 seconds; total CPU for the six jobs was 24.044414 seconds. Threaded solver work explains why CPU can exceed elapsed time. These are bounded scheduling problems, not six observed occupations or a claim to general project management.



Dark bars: J1 and J2 remain at start 0.
Native repair moves less than the retained candidate (20 versus 22).

Figure 3. The actual planning artifact changes under a longer job and then a capacity interruption. The final schedule preserves the two locked starts and avoids slots 8–10. These are supplied planning units; no physical work is claimed.

Table 3. Actual scheduling artifacts in the connected mission. Entries are start–end slots. Slots are supplied planning units, not measured employee hours. J1 and J2 remain locked in the repair.

Job	Initial plan	Changed plan	Selected repair
J1	0–2	0–2	0–2
J2	0–4	0–4	0–4
J3	8–10	9–11	6–8
J4	4–8	4–9	11–16
J5	10–13	11–14	16–19
J6	13–15	14–16	19–21
Finish slot	15	16	21

4.1 The successor consumes the actual parent artifact

The connected example is separate from those six primary jobs. The initial plan finished at slot 15. Increasing one integration duration produced a new, verified plan finishing at slot 16. Only after constructing and freezing the candidate methods did the worker evaluate them on the actual changed and disrupted service inputs. Development and qualification used the worker's fixed, disjoint variants; semantic collision checks could defer construction or a fresh trial rather than silently reuse an evaluated case. Inputs and custody remain visible, so this design does not claim a secret external holdout.

The subsequent disruption removed capacity in slots 8–10. The immediately preceding schedule required two crew units in each of those slots, making it infeasible under the changed calendar. The worker read that actual verified schedule and its receipt, reconstructed the baseline from its starts, and locked the jobs that had started before the supplied observation time. Independent acceptance derived the same baseline and locks from the original artifact and verified their receipt binding. The two locked starts remained unchanged. Table 3 shows the complete schedule rather than an abstract ancestry label.

The selected repair came from the native movement-first specialist: finish slot 21, total start-time movement 20. The constructed repair also finished at slot 21 but moved starts by 22 in total. Under the frozen objective—qualification first, then movement, then makespan for a qualified repair—the constructed candidate lost. The complete system delivered the better available result. Jobs described as completed or in progress are inferred from a planning observation, not physical execution; in this example the locked jobs were in progress at observation slot 1.

The complete journey took 8.096995 seconds of outer turnaround and 16.437662 CPU seconds. Its protected worker accounts recorded 32,380 service units and 284,211 improvement units, including candidate attempts, checks, reconstruction diagnostics and retention. Six native calls reserved twelve solver seconds. Reserved solver allowance is not actual solver time, and the explicit work units are neither processor instructions nor currency. Authoring, employer attention, infrastructure price and full maintenance remain outside a measured monetary total.

4.2 Inheritance is demonstrated; its advantage is narrower

Dispatch, lookahead and repair strategy artifacts were retained. The lookahead constructor read its dispatch parent and recorded 15 inherited calls; the repair constructor read the lookahead parent and recorded three. These counts describe actual construction execution, not merely matching digest fields. The full trial records preserve the candidate grammar, parent identity, rejected alternatives, qualification and source bindings. Retention does not automatically grant

Table 4. Matched parent-construction diagnostics. Values are the existing worker's declared work units, including parent preparation and child construction; they are not CPU seconds or monetary costs. All six child treatments qualified.

Child method	Inherited	Rebuilt	Equivalent
Lookahead	27,542	58,923	27,529
Repair	18,623	77,541	18,614

proof, production authority or economic admission: the capabilities remain held because full costs are unknown.

The worker additionally compared inherited, reconstructed and equivalent-parent treatments. Within each child stage, treatments received the same total allowance and child cap, with reconstruction, retrieval and unused allowances explicit. Table 4 reports their total declared work. Original parent acquisition is paid once in the complete journey; experimental rebuilding is an additional charged diagnostic. The equivalent treatment prevents a claim that only the particular retained representation can yield the child.

Inheritance avoided repeated parent construction. The equivalent recipe was slightly cheaper than the inherited representation on this declared accounting measure. This supports reuse as an engineering mechanism, while excluding a stronger claim of uniquely improved discovery or complete economic superiority. The costs of original acquisition, unsuccessful search and subsequent verification still matter when deciding whether a customer should pay for the method.

4.3 Fresh transfer keeps the specialist in the comparison

The retained repair was next offered on twelve fresh, four-to-six-job repair instances with different graphs, durations, baseline schedules and delayed input releases. Their historical schedules are authored task materials, not claimed observations of employee or physical work. Each input was evaluated twice, in randomized order: the complete ordinary/native worker, and that same worker with the retained strategy additionally available. Both arms kept the strongest ordinary and native tools. These are capability-mechanism comparisons; the paper's primary economic comparator remains traditional human work.

All 24 complete-worker calls were accepted, but the retained candidate lost to the strongest available method on all twelve instances. The product could therefore deliver useful work while the hypothesis that this particular learned repair improves end-use quality failed. Offering it increased the aggregate measured CPU ratio to 1.015389 relative to the ordinary/native arm. The paired, task-level bootstrap interval was [0.986398, 1.047411], using the frozen 2,000-resample analysis. This descriptive shared-host interval does not establish a speed improvement; complete monetary and attention effects remain unidentified. The original construction cost must also be paid before any later advantage could become net value.

The planning track consequently comprises six primary jobs, one connected journey and 24 transfer calls: 31 accepted operating attempts in total, not 31 independent planning task families. Total outer turnaround was 47.685766 seconds and recorded CPU was 136.713724 seconds. No confirmation retry or replacement was introduced. The actual requests, alternatives, checks, retained strategy, schedules and receipts are preserved in `raw/planning/`; the prespecified summary is `analysis/planning.json`.

Table 5. Bounded review confirmation. These are authored contract fixtures, not a natural prevalence estimate of professional document errors.

Condition	Cases	Observed outcome
Clean artifacts	6	No false alarms
Seeded contract defects	27	All detected
Fabricated criticisms	6	All rejected
Total registered attempts	39	All expected verdicts observed

5 Disagreement must survive an evidence check

Review has commercial value when the defects it prevents justify its complete cost. Additional discussion is not itself a useful output. We implemented the creator–challenger–adjudicator pattern as bounded, deterministic contract procedures for source-backed evidence artifacts. The creator assembles explicitly requested passages. The challenger identifies a violated requirement and provides the affected claim and expected or observed source binding. The adjudicator recomputes those facts before substantiating a criticism. Its verdict does not depend on a role name, vote count, or the confidence of an assertion.

This mechanism has a deliberately limited domain. It checks whether a required quotation is present, exact, correctly associated with a source and placed under the requested section. It also checks unrequested or duplicated claims and basic mission/title binding. It does not determine whether the underlying source is true, whether the employer selected the best evidence, or whether an unsupported free-form inference is scientifically persuasive. The critic and adjudicator share contract logic; their agreement cannot establish independent model judgment. A separately authored Python acceptance layer supplies an additional check of the frozen contract, while common research custody remains a limitation.

5.1 Registered defects, clean controls and false criticisms

The prospective experiment used six clean evidence inventories sampled from the frozen public source packets. Each contained three actual supplied passages with source hashes and URLs. We then created three instances of each of nine fault classes: missing required claim, incorrect quotation, incorrect source hash, incorrect URL, incorrect section, unrequested claim, duplicate claim, incorrect mission identifier and incorrect title. Six further cases supplied plausible but false criticisms against clean artifacts. An independent evaluator derived the expected findings directly from each brief and source packet. This produced 39 randomized review attempts, separate from the productive portfolio.

The critic detected all 27 seeded defects, produced no false alarms on the six ordinary clean cases, and the adjudicator rejected all six fabricated criticisms. Every expected verdict was observed; there were no interrupted or unknown review outcomes. Table 5 keeps the three denominators distinct. Counting the false assertions as criticisms generated by the system would misrepresent the test: they were authored adversarial inputs to adjudication.

Total measured review turnaround was 1.354492 seconds and combined child/supervisor CPU was 1.421349 seconds. The measurement includes process launch, actual review, independent result inspection and evidence writing through receipt creation; final receipt serialization is excluded. Each attempt had a ten-second process cap. The original artifact, mutation, criticism, adjudication and verdict are retained in `raw/review/` as reproducible regression cases. The finite pass demonstrates the registered checks' behavior; it cannot establish a general zero-error rate for arbitrary documents or the value of unlimited debate.

5.2 What these mechanisms permit us to claim

The planning example demonstrates useful delivery, an actual parent-dependent repair, retained executable strategies and fresh application. The review example demonstrates that unsupported criticism can be rejected through explicit evidence. Neither result licenses an inference from a larger archive to accelerating productive capability. The construction stages have different objectives and difficulty, full economic costs are incomplete, and the retained repair lost its fresh specialist comparisons. The acceleration scorecard must therefore remain `NOT ESTABLISHED`.

Operational scope is equally explicit. These are finite local executions of the existing fixed worker and workbench interface. They create no journal, credential, daemon, payment or production authority. An authenticated browser connection to an approved executor requires an already admitted operator environment and was not run here. The companion separates local computation from recorded evidence. The study's interruption demonstration is a bounded checkpoint interruption with an explicit new pure run, rather than continuation of an interrupted program counter; its local rollback restores a retained artifact selection, not a financial or physical transaction. Such distinctions protect the practical launch: the supported deliverables can be made available without claiming a continuously operating autonomous enterprise or hiding where a retained method failed.

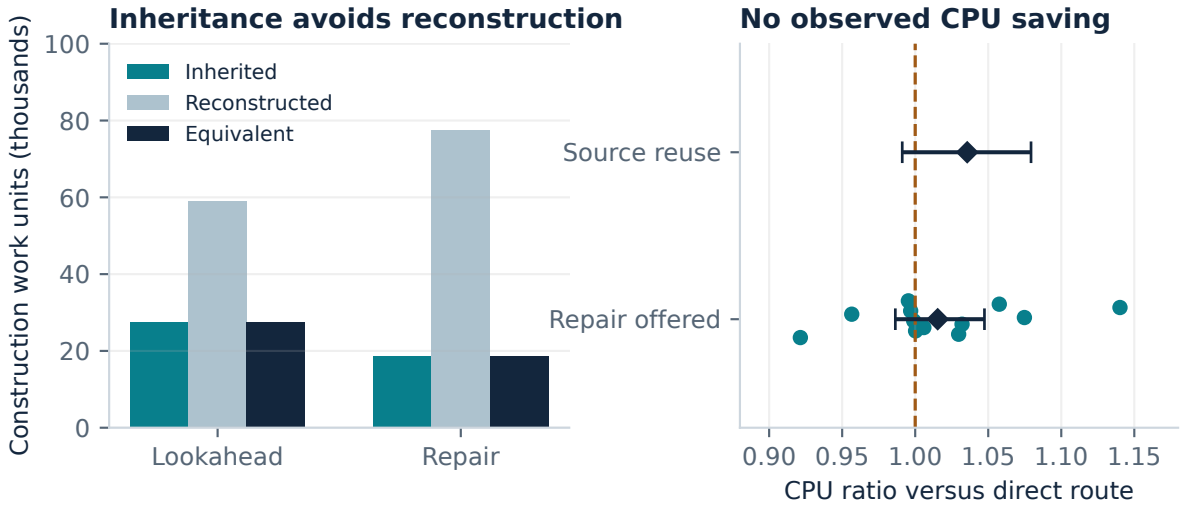


Figure 4. Useful inheritance and end-use advantage are different observations. Inherited parents avoid reconstruction units, while an equivalent recipe is as cheap. Neither the offered repair method nor source reuse demonstrates lower delivery CPU on the fresh comparisons. Full monetary costs remain unknown.

6 A second retained capability: maintaining source-backed deliverables

The new evidence adapter is an executable method for maintaining a constrained source-backed brief and deriving a presentation from it. It checks a bounded set of source texts and requested literal passages; creates the complete brief, source index, lineage and HTML presentation; challenges omissions, altered passages and source mismatches; and retains the executable repair module. This extends the available workbench interface to a supported task grammar that its default installed adapters did not accept. It creates a concrete new offering within this implementation. It does

not show that the same offering was previously impractical for every strong tool or competent worker under a matched resource budget.

The module was authored during study development. A mission retains its actual executable bytes; it does not invent that algorithm afresh. In a successor mission the worker validates the retained file against the installed source identity and then dynamically imports the parent artifact. A manifest alone is insufficient. The lineage records parent reads, raw and canonical parent hashes, changed source identifiers, preserved claims, regenerated claims and artifact-family transfer. Fresh acceptance never follows automatically from inherited provenance.

Table 6. The seven stages contribute concrete decisions and artifacts in the source-backed delivery loop. No financial execution is attached to these stages.

Stage	Inspectable contribution	Later use
Identify	Purpose, requested claims, source rights and input bounds.	Binds the external acceptance request.
Out-Learn	Validated source texts, URLs and content hashes.	Detects changed or removed evidence.
Out-Think	Claim-to-source dependency map and affected set.	Separates preserved from invalidated work.
Out-Design	Brief, source index and presentation schemas.	Produces usable artifacts with traceable claims.
Out-Strategise	Full or incremental route, bounded by measured admission.	Defaults to the full compiler when reuse fails the gain gate.
Out-Execute	Actual compilation, challenge, adjudication, export and external checks.	Delivers a newly accepted result.
Out-Evolve	Executable repair module, scope, failures and regression checks.	Invoked from the actual parent on fresh requests.

Twelve fresh change cases compare full compilation with incremental repair, randomizing the paired order. Twelve parent deliveries and twelve presentation transfers accompany the 24 paired attempts. Three subsequent policy validations each include a fresh parent and a fresh application, giving 54 accepted operating attempts in this track. A changed task alters the requested evidence inventory or uses another real pinned passage; it does not claim that the upstream repository policy itself changed during the experiment.

The first incremental case reads four parent claims, retains two unchanged claims and regenerates two affected claims. The second reads six parent claims, retains four and regenerates three, including a newly required passage. The independent oracle checks both the actual parent and the new result. All twelve cross-artifact applications also pass: the retained method supports a fresh presentation from the revised evidence. This is transfer between related artifact forms, not a new independent occupation or proof of general reasoning transfer.

The aggregate incremental/full CPU ratio is 1.0356, with a paired descriptive 95% bootstrap interval of [0.9911, 1.0793]. The result establishes actual reuse and traceable dependency repair, but no observed CPU saving. Tiny inputs make source loading, hashing, checking and process startup material relative to compilation. Whether a larger eligible workload changes the economics is a future hypothesis; these data are not reweighted to imply that result.

6.1 Can the system improve how it searches for useful methods?

Three prospective discovery cycles each examine four paired candidates with the same low/medium/high/medium complexity pattern and the same bounded search allocation. The policy is frozen and hashed before a fresh validation task. Incremental repair is admitted only

if all four paired delivery-CPU ratios are below 0.95 with no acceptance loss. This is a finite engineering gate, not a population-confidence test. Unknown full costs independently block economic admission.

The cycle median ratios are 1.0694, 1.0521 and 0.9971. No cycle meets the admission rule. All three policies retain full compilation and stop at the four-pair search budget. The paired search-evaluation CPU is 1.0492, 1.0872 and 1.1259 seconds, respectively, with all unsuccessful candidates charged. These totals cover the eight paired calls per cycle; parent deliveries and presentation transfers are additional recorded operations, not included in those three figures. A falling median over three small batches is not a verified gain, and a rising construction archive is not a rising improvement rate. There is one unique retained source-module implementation, not twelve discoveries counted from its copies.

For cycle k, define prospectively

$$G_k = \frac{\sum_{j \in \mathcal{F}} w_j \, \Delta V_{kj}^{\text{verified}}}{C_k^{\text{search}} + C_k^{\text{construction}} + C_k^{\text{qualification}} + C_k^{\text{deployment}} + C_k^{\text{human}} + C_k^{\text{maintenance}}}, \tag{1}$$

where $\mathcal{F}$, weights, acceptance and evaluation tasks are fixed outside the search. ΔV must represent verified useful capability or accepted-work improvement on fresh tasks under matched resources; it cannot count raw candidate volume. Self-acceleration requires an improving rate over successive cycles with complete costs, controlled difficulty and selection, and fresh transfer. Here three verified positive capability gains do not occur and the denominator is partly unmeasured. Self-acceleration is **NOT ESTABLISHED**.

7 Economics of accepted work and retained capability

An employer purchases an accepted result, its turnaround and the attention needed to obtain it. A retained method creates additional value when it improves later accepted work or makes a useful service feasible within a binding resource constraint. The relevant comparison is therefore the complete productive system against competent traditional human work using ordinary non-AI software. Human workers may use efficient workflows, templates and accumulated experience. The separate inheritance and specialist comparisons investigate mechanisms; they do not replace this primary economic comparator.

7.1 Observed labor prices, unobserved matched task times

The economic evidence combines official labor prices, occupational descriptions, published human-performance evidence and the study's separately reported local execution records. It is a market-data-grounded comparison, not a newly conducted human trial. The source matrix contains 24 price records, including two pairs of differently represented BLS wage estimates, rather than 24 independent surveys. It preserves occupation, geography, reference period, currency, benefits coverage and methodology. O*NET supports the relevance of regression testing, technical documentation and project-plan revision; it does not identify how much of those occupations the bounded implementations can perform.[25–27]

Wage-only prices receive explicit employer-load sensitivities of 1.00, 1.30 and 1.50. Benefits-inclusive ECEC and Eurostat prices receive no second benefit multiplier. ILOSTAT definitions help distinguish earnings from employer labor cost; currencies and survey scopes are preserved rather than combined into a fictional global hourly rate. The labor statistics price the stated non-AI comparator but do not establish that every sampled worker used no AI. [16, 20, 34]

Published timings remain adjacent evidence. Peng et al.'s professional-programmer control averaged 160.89 minutes among completers of a JavaScript HTTP-server task requiring twelve passing tests;

Table 7. Selected observed labor-price anchors. AI use among source workers is not established. None measures the duration or quality of the present missions.

Occupation or scope	Period	Currency/hour	Boundary
U.S. software developer	May 2025	USD 65.375	Wage only[a]
U.S. office/admin	Dec. 2025	USD 35.96	Wage + benefits
U.S. professional/related	Dec. 2025	USD 72.19	Wage + benefits
Quebec software developer	2023–2024	CAD 45.67	Wage only
Ontario technical writer	2023–2024	CAD 34.98	Wage only
EU covered economy	2025	EUR 34.90	Employer labor cost

[a]Annual median USD 135,980 divided by an assumed 2,080 hours; this conversion is not observed working time. BLS ECEC figures are broad-group means. Canadian prices are provincial Labour Force Survey medians distributed by Job Bank, updated November 2025. Eurostat covers specified sectors, normally enterprises with at least ten employees, and is not occupation-matched. [14–16, 34, 35]

repository timestamps defined elapsed time. Controls lacked Copilot, but the source does not establish exclusion of every AI tool. This is not a duration for the present utility repairs.[31] The March 2023 Noy–Zhang working paper reports a 27-minute self-reported control mean and ChatGPT use by 10–20% of controls; its time is excluded from strict non-AI calibration.[29] Earlier manual transformation evidence also involves different tasks and censored failures.[13] Consequently, no borrowed duration becomes a matched human denominator here.

7.2 Cost, attention and the adoption threshold

The operating records contain 42 accepted primary deliveries from 42 attempts. For the 36 non-planning jobs, producer launch through independent acceptance and export took a median 0.112 seconds (range 0.091–0.263). The six planning jobs' complete child-process boundary took a median 1.301 seconds (1.216–1.337). These boundaries are reported separately and exclude development and employer attention. Repeated twelve-job non-planning batches had median elapsed times of 1.837, 0.996 and 0.578 seconds at concurrency one, two and four; all 108 attempts passed. This measures finite local delivery, not sustained service or human time displacement. The operational observations therefore join the economic model without filling its missing cost and attention fields.

An observed cost per accepted deliverable divides all attributable delivery and allocated improvement costs by accepted outputs. Rejections, retries and abandoned attempts remain in the numerator. For a conditional repeat-until-accepted model, let t be traditional human minutes per attempt, h remaining human minutes in the Agent workflow, w_H, w_A their hourly employer prices, v_H other human-arm cost and k all other allocated Agent cost. With acceptance probabilities q_H, q_A,

$$C_H = \frac{w_H t/60 + v_H}{q_H}, \qquad C_A = \frac{w_A h/60 + k}{q_A}. \tag{2}$$

Effort includes execution, supervision, review and correction. The Agent's k must include provider usage, compute, setup, verification, failed attempts where not already represented by q_A, construction, deployment, support and maintenance. Charging failures both in k and again through acceptance would double count them. Whole-process wall time and CPU are operational measurements; neither identifies complete project monetary cost or employer attention. Material authoring, attention and maintenance costs remain unknown in this study.

For equal acceptance, equal labor prices and $v_H = 0$, break-even traditional effort is $t^* = h + 60k/w$. At the observed U.S. office/admin price of USD 35.96/hour, *assumed* $q_H = q_A = 1$, $h = 5$ minutes and $k =$ USD 1 imply Agent cost USD 3.997 and break-even at 6.669 traditional minutes. An assumed 30-minute traditional workflow costs USD 17.98, a conditional 4.50-fold cost ratio. A

Table 8. Explicit 2025 U.S. scenarios, in current USD trillions per year. Digital-work shares are assumptions; these are not measured addressable markets.

Scenario	Eligible share	Computer share[a]	Supplied base	BEA base
Conservative	70.0%	45%	4.9455	4.9515
Central	77.5%	60%	7.3005	7.3093
Ambitious	85.0%	75%	10.00875	10.0208

[a]Within eligible work. Multiply each share pair by USD 15.70 trillion for the supplied scenario or USD 15.718937 trillion for the refreshed domestic BEA denominator. Complete precision is retained in the evidence files.

five-minute traditional workflow costs USD 2.997 and wins instead. The professional-group price yields Agent cost USD 7.016 and break-even at 5.831 minutes under the same assumptions. These calculations are inspectable scenarios, not measured wins.

The frozen grid contains 810 combinations of two employer prices, five human durations, three attention durations, three other-cost values and independent acceptance assumptions. In 266 combinations, modeled Agent economic value is nonpositive. This unweighted count is sensitivity, not an empirical probability. The six-fold attention-capacity ratio in the 30-versus-five-minute example is likewise conditional; actual capacity also depends on accepted executor throughput, demand and the available budget. Released employee time is not automatically a payroll saving, provider revenue or cash for reinvestment.

7.3 National scale and a bottom-up service model

BEA's September 2026 vintage reports 2025 domestic employee compensation paid of USD 15.718937 trillion. This includes wages and employer supplements for domestic production, including payments to nonresidents, and excludes compensation received from abroad and proprietors' income. It is neither GDP nor wages alone. [32, 33] Table 8 preserves the supplied USD 15.70 trillion rounded-base scenario and separately applies the same assumed shares to the refreshed denominator. Neither occupation eligibility nor computer-work share is an official statistic or an inference from fixture acceptance.

A separate bottom-up model assigns disjoint task shares within ten distinct BLS occupations, then family-specific matching-context and adoption shares. Employment times median annual wage is explicitly a standardized wage-value proxy, not payroll: medians differ from means, and employment includes self-employment while the wage survey excludes it.[35] All allocation and context fractions are prospective assumptions, not estimated from this portfolio. In the central scenario, USD 46.19 billion of relevant task proxy narrows to USD 19.49 billion in assumed matching contexts and USD 3.90 billion after 20% adoption. An assumed 20% service fee yields USD 779.59 million for all providers; an assumed 1% obtainable provider share yields USD 7.80 million. None is actual revenue. The cohort proxy cannot be added to national compensation or savings.

Separate 2026 sensitivities apply assumed nominal growth of 0%, 4% or 8%; they are not annual observations or official forecasts. The global USD 40 trillion thesis remains an illustrative scenario. ILO's estimated 2025 labor-income share of 52.6% of GDP does not identify computer-mediated employer expenditure. [19] Language, infrastructure, procurement, reliability and integration constrain serviceability beyond the tested tasks.

7.4 Paying for improvement

Accepted delivery did not automatically produce cheaper improvement. The inherited evidence path used 1.036 times the full compiler's CPU, and all three discovery cycles retained full

compilation. In planning transfer, the retained repair candidate lost to an available specialist on all twelve cases, although the complete product accepted all 24 paired outputs. Offering the retained candidate produced a whole-worker CPU ratio of 1.015 (descriptive bootstrap interval 0.986–1.047). These measured outcomes support useful delivery while withholding a cost-saving claim for these retained methods. Complete monetary cost and human attention remain unknown; the CPU ratios are not dollar ratios.

Separate delivery and improvement accounts must be combined before claiming net value. With full construction cost K and positive net saving ΔC per accepted reuse, payback requires $N^* = \lceil K/\Delta C \rceil$ accepted reuses. At $K = 100$ currency units and $\Delta C = 1$, the threshold is 100; at $\Delta C = 0.01$, it is 10,000. Unsuccessful search, evaluation and deployment belong in K; verification, regressions and maintenance reduce ΔC. Nonpositive saving has no finite cost payback absent another verified benefit.

The frozen eight-cycle accounting simulation varies demand, price, reliability, utilization and diminishing returns. Only 63 of 243 scenarios permit any reinvestment under its finite positive-margin allowance. No money is received or spent. These paths clarify accounting; they establish neither actual margin nor self-acceleration. Demonstrated reuse, better method selection and expansion of a supported task grammar remain distinct from a rising rate of verified gain per complete resource cost.

8 Interfaces to larger productive systems

The digital mechanisms suggest concrete next interfaces: evidence packets and numerical checks to laboratory records; requirements and dependency repair to CAD/CAE and test plans; quality records and versioned procedures to manufacturing systems; and constraint checks to infrastructure procurement and schedules. These are proposed engineering pathways, not physical productivity findings. NASA's systems-engineering distinction between requirements verification and suitability validation explains why a correct digital artifact is insufficient. [24] NIST's manufacturing digital-twin work similarly requires traceability, interfaces, validation and uncertainty treatment. [28]

Scientific results still require calibrated instruments, materials, domain judgment and reproducible physical experiments. Engineering requires appropriate physical tests and manufacturing tolerances. Industrial deployment needs trusted sensors, machines, inventory, logistics and accountable operating decisions. Infrastructure adds permits, capital, energy supply, construction capacity and equipment lead times. Compute itself requires servers, electricity, cooling and network capacity. The IEA estimates global data-centre electricity use of 415 TWh in 2024 and projects 945 TWh in its 2030 base case, including non-AI data centres; it reports transmission construction taking 4–8 years in advanced economies. [18] These contextual figures are not Alpha task measurements. Faster digital work may improve the use of scarce resources, but inexpensive software replication does not supply those physical resources or establish unlimited productive growth.

9 Operating capacity, intervention and the browser companion

The scale experiment uses twelve existing low/high non-planning instances, two per eligible family, in three finite batches at each concurrency of 1, 2 and 4. Batch order is randomized. Every job runs in its own process so cumulative child CPU is not incorrectly attributed across overlapping jobs. All 108 attempts are accepted; these repetitions add no task diversity. Each batch begins with twelve submitted jobs, and queue time is included in submission-to-delivery latency.

Median accepted throughput is 391.91, 722.96 and 1,244.71 outputs per minute at concurrency 1, 2 and 4, respectively. The four-to-one ratio is 3.176. The medians of the twelve-job completion

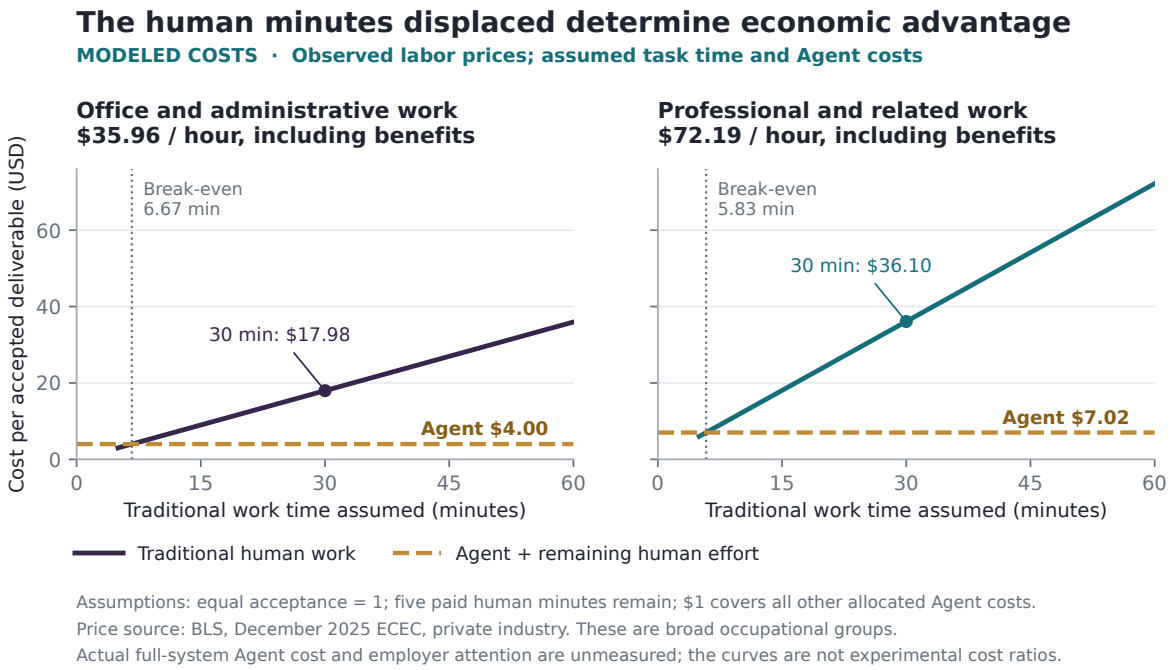


Figure 5. Conditional employer economics. Official labor-price observations are combined with explicitly assumed human duration, remaining attention and other Agent costs. These are break-even and sensitivity calculations, not observed Alpha savings or newly measured human performance.

intervals are only 1.837, 0.996 and 0.578 seconds. These rates normalize tiny finite batches to a minute; they do not establish sustained minute-long throughput, a customer arrival process, an enterprise fleet or capacity for the planning/native-solver family. Median submission-to-delivery latencies are 1.017, 0.560 and 0.377 seconds. Aggregate CPU rises from 6.188 seconds across the concurrency-one batches to 7.001 seconds across the concurrency-four batches. Faster elapsed completion is not lower total resource use.

9.1 Recovery and refusal are operating results with defined scope

All sixteen registered controls pass: pre-execution STOP, injected checkpoint interruption, a simulated-clock time limit, explicit fresh restart, eight unsupported or restricted request classes, retained-code corruption, installed source mismatch, immutable-output protection and local pointer rollback. The restart is a new pure operation; it does not establish resumption of a deployed daemon. The rollback restores an artifact pointer without modifying the parent bytes; it is not reversal of a financial or external business transaction.

The refused classes are arbitrary software repair, unstructured summarization, financial execution, physical action, private-login work, clinical prediction, restricted personal data and inputs beyond the declared bound. Refusal is counted as successful containment and separately from delivered work. Dollar-budget enforcement is **NOT RUN**: the worker has no authorized spending route. The existing worker's finite work allocations and execution bounds are real local controls, not evidence of monetary-budget management.

Intervention records preserve setup, source adjustments before freeze, the two earlier sandbox-denied child-process launches, the approved execution path, validation and publication work. No confirmation output required human rescue or rework. This does not mean zero human effort: employer attention was not timed, and authoring, supervision, infrastructure administration and support are not fully billed. The study clock is not a human timesheet. The delivery and improvement ledgers expose this incompleteness rather than amortizing it away.

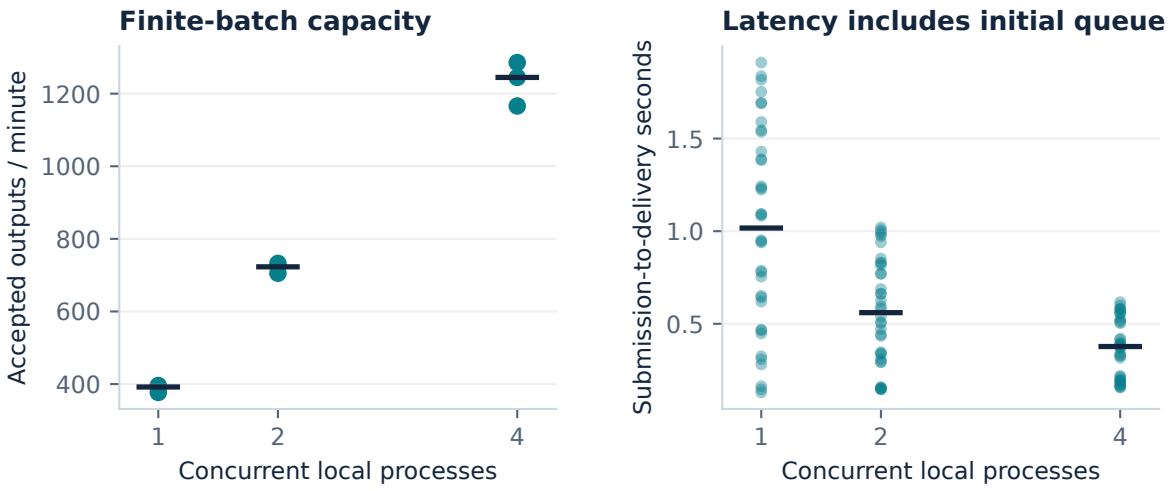


Figure 6. Measured finite local load. Throughput shows all three batches at each setting; latency shows all 36 deliveries at each setting. The experiment has a finite initial backlog and no steady-state arrival stream. Supervision and host price remain unmeasured.

9.2 A browser product with observable states

The companion begins with a useful request, displays the delivered brief and source links, then shows a changed mission and the retained executable's actual contribution. The live path runs the same frozen evidence module through the maintained workbench inside the browser. It can export the evidence package and show STOP and local rollback. The planning journey is a labeled recorded replay of actual worker outputs; the economics controls are a labeled simulation. The approved-executor panel explains the existing operator route and its required authority, while reporting **NOT RUN** for this study. No new credential or unauthorized executor is created to turn that label green.

The page communicates five questions directly: what result arrived, what cost is known or unknown, what was retained, how the successor used it, and which bounded capability is now available. Closing the browser ends browser-local computation. An independently launched local worker has only the lifetime of its active process; no background service was installed. No continuously operating company is implied by a finite foreground demonstration.

10 Mechanisms inherited from the research program

The foundational AGI ALPHA architecture connects delivered work, an evidence docket and reusable productive capability. TRANSFORMATIONAL INTELLIGENCE separates purpose and evaluation from the method chosen to satisfy them. These ideas motivate a fixed employer brief with flexible legitimate execution methods. They do not substitute architectural ambition for measured performance.[1, 12]

The Out-stage papers supply concrete design disciplines: acquire useful observations, preserve relevant distinctions in representations, construct replacements without losing service, allocate observation and construction under uncertainty, maintain valid execution progress, and build the means of future discovery. Table 6 shows their actual contributions to this implementation.[4–9] The Possibility Foundry motivates explicit resource frontiers, complete construction accounting and preservation of stronger alternatives and negative results.[10]

Mission-Sovereign Successor, neural-symbolic SUCCESSOR Ω, Mission Gym and SUCCESSOR Ω × AGI Jobs motivate scoped authority, executable strategies, fresh falsification, mission acceptance and typed evidence dependencies.[2, 3, 11, 23] The original Jobs compiler produces seven typed, unfunded work packages from the actual stage artifacts. Their proof and authority fields remain empty. They are drafts, not contracts executed on a chain.

The broader literature shows that model-written critiques can assist error finding and that evaluator-guided search can discover useful algorithms. [17, 30] Those results motivate investigation; they do not establish that our deterministic challenger is an independent model judge, or that our bounded search has reproduced open-ended algorithm discovery. Our contribution is the connected artifact lineage, separately checked delivery, strong-tool fallback and explicit economic admission around these mechanisms.

The four-repository source audit distinguishes AGI ALPHA, AGIJobManager, AGIJobs v0 and AGI-Fleet-Linux. Manager's USDC commissioning route is separate from the $AGIALPHA routes. Five Alpha contracts—AscensionAccess, NovaSeed, AscensionRiskOracle, AscensionJobMarket and AscensionMark—have relevant source-level eligibility, plan, review, commissioning and routing functions. The retained audit is a historical source inspection, not a fresh deployed-state qualification. No transaction, wallet funding, stake, mint, settlement or deployed-contract modification occurred. Existing holds and release-owner boundaries remain effective.

11 What can be launched, and what remains a research claim?

The supported product is a bounded local workbench that delivers checkable files, retains executable methods and provenance, refuses incompatible requests and keeps a strong specialist available. This supports a focused launch of the shareable research companion and narrowly specified local workflows through the existing review process. It does not require a universal-agent claim or a positive self-acceleration result. Production admission requires the approved operator route, complete cost observation, target-user acceptance and the repository's existing release gates.

The most decision-relevant negative finding is that a retained method need not deserve deployment. Inheritance can save reconstruction effort while the constructed method remains worse than an available specialist. A productive system must be able to retain a reusable artifact for study, reject its economic admission, and still deliver the customer's work through the better route. That behavior protects the commercial objective from becoming a demand to prove recursion.

The next empirical priority is complete observation at a real approved deployment boundary: actual employer input preparation, review, corrections, support and maintenance, priced infrastructure, and comparable traditional workflow records. Existing authorized employer records or published matched task evidence can supply the human benchmark; a new recruited panel is not a prerequisite. Further capability research should target workloads where dependency reuse is a material share of total cost, preserve strong direct alternatives, and freeze independent acceptance before new confirmation. A broad economic opportunity is a reason to make these tests rigorous, not a reason to extrapolate this selected portfolio.

Table 9. Launch scorecard. Observations, modeled economics and unestablished claims remain separate.

Question	Evidence	Status
Useful delivery	42/42 primary jobs accepted by frozen independent implementations.	Observed locally
Breadth	Seven bounded families; eight templates; two complexity instances per stratum.	Scoped coverage
Inheritance	Actual parent reads and 15/3 inherited calls; reduced reconstruction work.	Demonstrated
Capability transfer	Source module reused on changed briefs and presentations; repair candidate loses 12/12 to strong alternatives.	Mixed; loss retained
Discovery efficiency	Three four-pair cycles, no admitted CPU improvement; stop rule obeyed.	No gain established
Review reliability	27/27 seeded defects detected; six clean cases without alarm; six false criticisms rejected.	Bounded checks
Operating capacity	108/108 finite-batch deliveries; 3.176 times median throughput at four versus one process.	Local finite load
Net economics	Official prices and conditional break-even analysis; actual full cost, attention and matched human duration unknown.	Conditional model
Acceleration	No three positive complete-cost capability gains or increasing verified rate.	Not established
Authenticated service	Existing approved executor not invoked; no service or credentials created.	Not run

A Protocol and evidence map

The prospective issue notice is issue #4850, comment 6397672653. The checkpoint ZIP SHA-256 is

`67a793b9ed781aa62ae9ce1469c701adc66c3149f3a36631812263aa8b59e4d5.`

The repository source pin, file-level freeze and output inventories are supplied in the companion. Public source notice, draft review, merge, release and live-site verification are separate states; their current values appear in `publication/STATUS.json`. A public research draft is not a production release or a clearance of an existing hold.

Table 10. Companion navigation. All paths are relative to the study root.

Path	Contents
`protocol/PROTOCOL.md`	Primary and mechanism questions, limits, acceptance, analysis, exclusions, controls and prospective nonclaims.
`protocol/FREEZE.json`	Exact identities of 110 frozen inputs.
`raw/portfolio/`	Thirty-six non-planning requests, producer outputs, external checks and full local receipts.
`raw/planning/`	Six primary jobs, connected journey, twelve paired transfer instances, actual schedules, construction, parent diagnostics and cost accounts.
`raw/evidence-mechanism/`	Twelve parent/change/presentation sequences, paired routes and three frozen-policy validations; 54 operating attempts.
`raw/review/`	Thirty-nine clean, seeded-defect and false-criticism cases.
`raw/controls/`	Sixteen STOP, interruption, restart, refusal, source-integrity and rollback records.
`raw/scale/`	Nine finite load batches, 108 delivery records on twelve inputs.
`analysis/`	Frozen aggregate calculation, planning analysis, scorecard and transparent delivery/improvement ledgers.
`economics/`	Twenty-four labor-price records, source methods, family mapping, sensitivity, bottom-up scenario model and bounded reinvestment simulation.
`development/`	Development attempts and failures, earlier uninformative planning case, repairs made before freeze and evaluator mutation checks.
`evaluation/`	Independent acceptance implementations and integrity audits.
`upstream/` and `strategy_worker/`	Pinned original runtime modules and source manifests.
`src/`	Frozen study adapter, wrappers, generators and analysis.
`browser/`	Accessible companion, local computation, replay, simulation and downloadable evidence.
`paper/` and `figures/`	Compilable LaTeX, bibliography, clean build recipe and editable original vector figures.
`reproduction/`	Fresh-output execution and audit instructions; no credential creation or financial execution.

A.1 Predeclared seeds and denominators

The primary non-planning seed is 2026102507; planning is 2026102517; source mechanisms are 2026102527; review is 2026102537; controls are 2026102547; bootstrap analysis is 2026102557. Load reuses the primary non-planning seed on the declared twelve-input subset. Each planning

transfer case contributes one paired observation, not its internal candidate trials. The connected three-phase journey is one operating attempt and is not added to the 42 primary jobs. The 39 review audits and 16 control cases contribute zero productive-job observations.

The full raw record keeps every attempted operating path. Confirmation contains no operating retries or abandoned cases. Development failures remain separate, and candidate losses remain visible inside accepted complete-product outputs. Counting only accepted product outputs without their failed internal candidates would conceal improvement cost; counting every candidate as another useful job would inflate productive capacity. Neither is done here.

B Reproduction and accounting boundaries

The reference environment uses Python 3.12.14, Node 24.22.1, SciPy 1.17.0, NumPy 2.4.6, Pydantic 2.13.5 and cryptography 50.0.1. Plot generation uses Matplotlib 3.10.8. The exact source and environment manifests accompany the study. Local native subprocesses require the execution environment to permit the existing worker's Node-to-Python call; the preserved development record contains the sandbox failure when that capability was absent. Reproduction must use an approved environment rather than alter that boundary.

Use a fresh extracted copy for reruns; immutable-output protection deliberately refuses existing result directories. Verify the input freeze before running the documented commands. The reproduction script writes to new paths and leaves the published raw evidence intact. Pure task execution has no required network, wallet or newly created credential. Market sources can be refreshed separately, but doing so creates a new economic vintage and does not modify this frozen study.

The delivery ledger measures local CPU and turnaround. The improvement ledger includes all observed candidate construction, qualification, parent reconstruction, paired search and failed development where metered. Some receipts combine delivery and improvement in one process; the worker's service/improvement units separate logical accounts, while CPU is kept joint rather than divided by an invented ratio. Initial authorship, full setup, employer time, support and maintenance remain explicit unknowns. Task-time zero API calls do not imply zero research-model consumption.

The full accepted-deliverable cost is identifiable only when all material cost components and acceptance failures are accounted for. Amortization needs a declared useful lifetime, expected eligible reuse, support burden and failure rate. A retained artifact with no verified net saving has no demonstrated break-even reuse volume. The scenario model illustrates how positive future savings could cover construction; it does not turn observed CPU into cash.

C Validity limits and claim boundaries

Selected breadth. Seven families span recognizable digital-work categories, but each has a narrow grammar and small instances. Complexity labels are within-family definitions. The portfolio is not a random labor-market sample. Refused categories and untested responsibilities are part of the coverage map.

Task and evaluator knowledge. The team authored both task generators and implementations before freeze. Fresh seeds prevent repeated confirmation tuning but do not create an unseen domain. Separate evaluator implementation and semantic mutation checks reduce some errors without eliminating common conceptual mistakes. There is no blinded employer-quality panel.

Timing. Very short tasks are sensitive to process startup, filesystem caching and host scheduling. All attempts and raw distributions are retained. Small bootstrap intervals describe this sample,

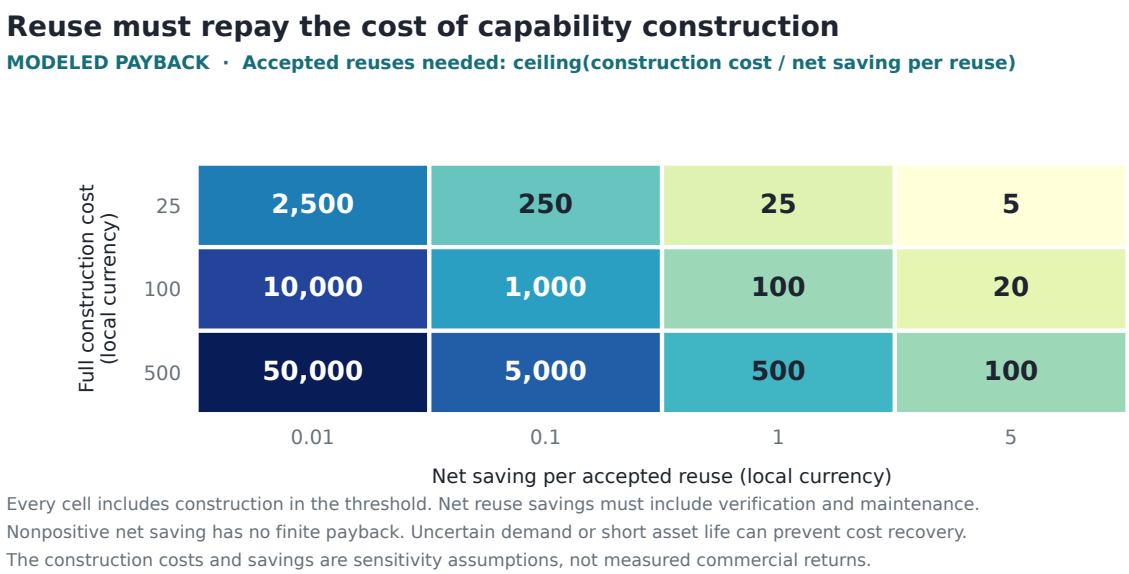


Figure 7. Modeled construction payback. Reuse counts depend on assumed positive net savings per use and construction cost. An ineligible task, a losing retained method or nonpositive net saving does not gain a finite payback count by being repeated.

not a different machine or remote deployment. Native solver CPU includes threads and is not interchangeable with elapsed time or paid core allocation.

Economics. Wages do not establish task duration, worker quality or benefits. Official employer costs still need task-specific attention and correction measurements. Published control groups may allow AI or have selection and completion censoring. The scenario range is not a confidence interval. Released capacity is not necessarily a reduced salary, customer savings are not provider revenue, and accounting margin is not automatically spendable cash.

Learning. Actual inheritance, avoided reconstruction, artifact-form transfer, improved output quality, lower complete cost and an accelerating rate of improvement are distinct. The present study demonstrates the first three in bounded forms, retains negative findings on the next two, and does not establish the last. No count of agents, archive size or rhetorical stage label changes that conclusion.

Operations and authority. Source inspection is not production execution. Local rollback is not external transaction rollback. Finite worker bounds are not spending controls. A browser replay is not a live executor. The USDC and $AGIALPHA routes remain distinct and unfunded. No participant identity, private employer content or credential is included in the public materials.

Market scale. The supplied U.S. $5–10 trillion range is a scenario constructed from 2025 compensation and assumed digital-work shares. Separate 2026 growth assumptions and the illustrative global $40 trillion hypothesis are not official addressable-market statistics. Neither the 42/42 acceptance fraction nor a local throughput ratio is applied across those totals.

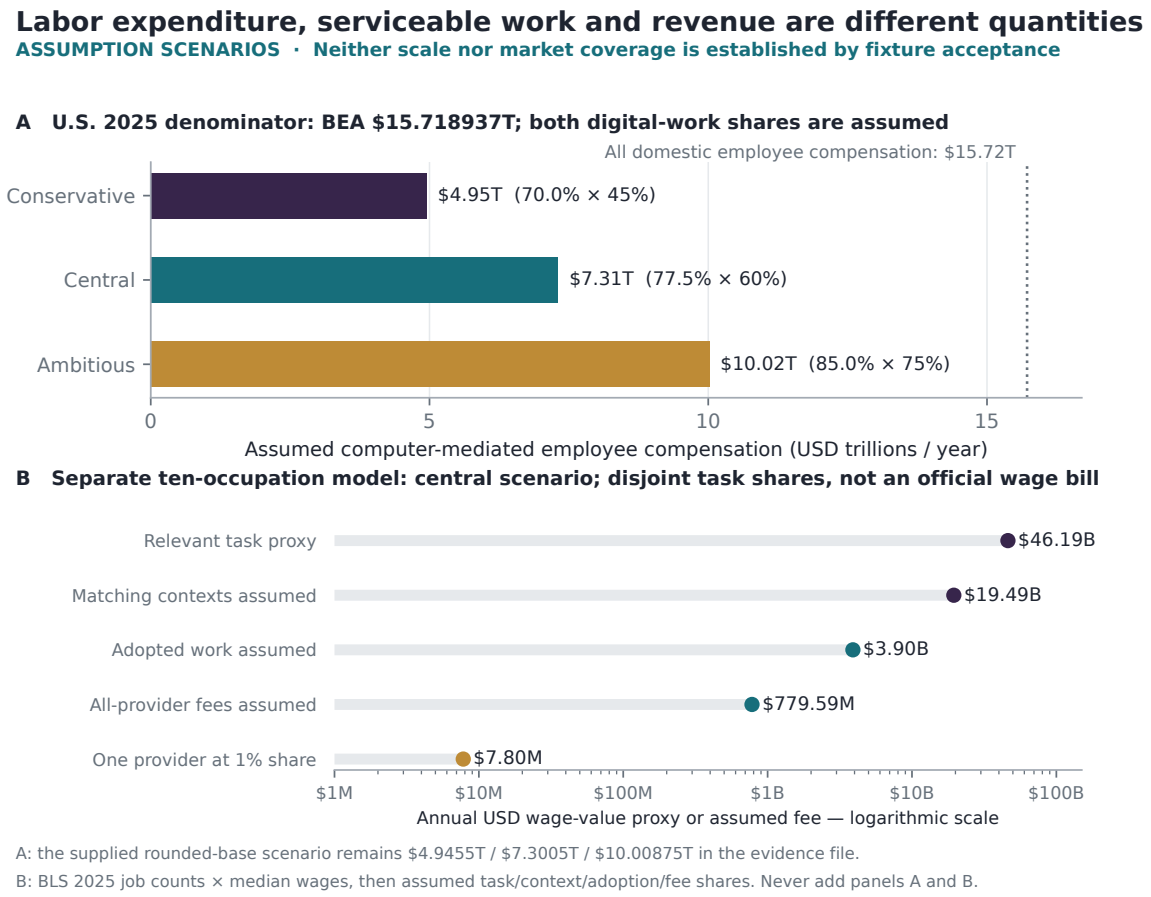


Figure 8. Different economic scopes. Official total compensation, an assumed digital-work share, bottom-up standardized wage-value scenarios, serviceability and obtainable provider revenue are separate quantities. The experiment measures none of the assumed global or occupation-wide automation shares.

References

[1] Vincent Boucher. AGI ALPHA: A scalable substrate for intelligence organizations. MONTREAL.AI research manuscript, 2026. URL https://github.com/MontrealAI/AGI-Alpha-Agent-v0/blob/f836f06e8351e64db1da770e16e304b5feb01f28/docs/manuscript/AGI_ALPHA_Unified_Publication_Final.md. Original Markdown inspected, 25 October 2026; architecture and bounded local evidence.

[2] Vincent Boucher. Gym, specialist asi and successor omega: The executable succession institution, August 2026. URL https://montrealai.github.io/Gym_Specialist_ASI_Successor_Omega_v2_1_0_Print.pdf. Version 2.1.0, 10 August 2026; PDF SHA-256 73bad8930e02e5c81e59add0a029c76b3f1b0f9b4f54cf17dfcf2b2768cfb7d8.

[3] Vincent Boucher. Mission-sovereign successors: Goalos singularity navigator omega + seize, August 2026. URL https://montrealai.github.io/goalos-singularity-navigator-omega-seize-agi-alpha-or-agi-club-owner-access-v1/research/GoalOS_Singularity_Navigator_Omega_SEIZE_Mission_Sovereign_Successors_v5_0_0_A4.pdf. Version 5.0.0, 9 August 2026; PDF SHA-256 9dd9caa6170bd1905e52926516d687c47313d58cd5e2a59e6038eabf0014639b.

[4] Vincent Boucher. Out-design, 2026. URL https://montrealai.github.io/AGI-Alpha-Agent-v0/Out_Design.pdf. Executable release certificates and usable transition capacity; original LaTeX inspected.

[5] Vincent Boucher. Out-evolve: Building the means of discovery and production, October 2026. URL https://montrealai.github.io/AGI-Alpha-Agent-v0/Out_Evolve.pdf. 20 October revised research manuscript; PDF SHA-256 5979bcaf704934ba42e4722389cbb1a9ef0cbc889e1b286428615a82a31e99bc.

[6] Vincent Boucher. Out-execute: Constructing the conditions of completion, 2026. URL https://montrealai.github.io/AGI-Alpha-Agent-v0/Out_Execute.pdf. Boundary contracts, fresh acceptance and recovery; original LaTeX inspected.

[7] Vincent Boucher. Out-learn, 2026. URL https://montrealai.github.io/AGI-Alpha-Agent-v0/Out_Learn.pdf. Revised experiment-construction and calibration study; original LaTeX inspected.

[8] Vincent Boucher. Out-strategise, 2026. URL https://montrealai.github.io/AGI-Alpha-Agent-v0/Out_Strategise.pdf. Joint observation and construction under uncertainty; original LaTeX inspected.

[9] Vincent Boucher. Out-think, 2026. URL https://montrealai.github.io/AGI-Alpha-Agent-v0/Out_Think.pdf. Purpose-preserving executable representations; original LaTeX inspected.

[10] Vincent Boucher. The possibility foundry, October 2026. URL https://montrealai.github.io/AGI-Alpha-Agent-v0/papers/possibility-foundry-v1.pdf. 23 October executable digital laboratory; original LaTeX inspected.

[11] Vincent Boucher. Successor omega x agi jobs: The constitutional compiler and proof-carrying work system for mission-sovereign intelligence institutions, August 2026. URL https://montrealai.github.io/successor-omega-x-agi-jobs-v2.pdf. Version 2.0.0, 12 August 2026; PDF SHA-256 c25b911f20ac623bdb26f520376f2b1f0e774eba982db1c103a5e73d4b1d1a0d.

[12] Vincent Boucher. Transformational intelligence: Learning to change the game, 2026. URL https://montrealai.github.io/AGI-Alpha-Agent-v0/Transformational_Intelligence.pdf. Original LaTeX inspected at Alpha commit f836f06e8351e64db1da770e16e304b5feb01f28.

[13] Ian Drosos, Titus Barik, Philip J. Guo, Robert DeLine, and Sumit Gulwani. Wrex: A unified programming-by-example interaction for synthesizing readable code for data scientists. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020. doi: 10.1145/3313831.3376442. URL https://pg.ucsd.edu/publications/Wrex-synthesizing-readable-data-science-code_CHI-2020.pdf.

[14] Employment and Social Development Canada. Job bank: Technical writer in ontario, wages, 2025. URL https://www.jobbank.gc.ca/marketreport/wages-occupation/20923/ON. NOC 51112; Statistics Canada LFS 2023–2024; median CAD34.98/hour; updated 19 November 2025; accessed 25 October 2026.

[15] Employment and Social Development Canada. Job bank: Software developer in quebec, wages, 2025. URL https://www.jobbank.gc.ca/marketreport/wages-occupation/22548/QC. NOC 21232; Statistics Canada LFS 2023–2024; median CAD45.67/hour; updated 19 November 2025; accessed 25 October 2026.

[16] Eurostat. Eu hourly labour costs ranged from 12 to 57 euros in 2025, March 2026. URL https://ec.europa.eu/eurostat/en/web/products-eurostat-news/w/ddn-20260331-2. Published 31 March 2026. Dataset lc_lci_lev; enterprises with 10 or more employees, covered economy. Accessed 25 October 2026.

[17] Google DeepMind. Alphaevolve: A gemini-powered coding agent for designing advanced algorithms, 2025. URL https://deepmind.google/blog/alphaevolve-a-gemini-powered-coding-agent-for-designing-advanced-algorithms/. 14 May 2025; evaluator-guided algorithm discovery, not evidence of AGI ALPHA performance; accessed 25 October 2026.

[18] International Energy Agency. Energy and ai: Executive summary, 2025. URL https://www.iea.org/reports/energy-and-ai/executive-summary. Estimated 2024 data-centre electricity415 TWh; 2030 base-case projection945 TWh; grid/lead-time constraints. Accessed 25 October 2026.

[19] International Labour Organization. The world of work and the 2030 agenda: a ten-year review, 2026. URL https://ilostat.ilo.org/blog/the-world-of-work-and-the-2030-agenda-a-ten-year-review/. 19 March 2026; 2025 global labor-income share 52.6 percent; not an employer-cost total. Accessed 25 October 2026.

[20] International Labour Organization. Ilostat: Statistics on earnings and labour income, 2026. URL https://ilostat.ilo.org/topics/wages/. Methods, source-country scope and local-currency concepts; accessed 25 October 2026.

[21] MONTREAL.AI. Strategy succession: Fixed worker, construction, native specialist and transfer, 2026. URL https://github.com/MontrealAI/AGI-Alpha-Agent-v0/tree/f836f06e8351e64db1da770e16e304b5feb01f28/contrib/strategy_succession. Exact selected source files pinned and independently verified for this study; accessed 25 October 2026.

[22] MONTREAL.AI. Successor omega: Useful local work, 2026. URL https://github.com/MontrealAI/AGI-Alpha-Agent-v0/blob/f836f06e8351e64db1da770e16e304b5feb01f28/docs/agent/SUCCESSOR_WORK.md. Source and implementation inspected 25 October 2026.

[23] MONTREAL.AI. Neural-symbolic successor omega: Customer-owned mission intelligence, August 2026. URL https://montrealai.github.io/neuralsymbolicsuccessoromegav1.pdf. Definitive corporate white paper, 25 August 2026; PDF SHA-256 36e7c26fe28961dbd8a0dc89160a0462e08acb53fc9b00e2508a8740b52847e9.

[24] National Aeronautics and Space Administration. Systems engineering handbook: Fundamentals of systems engineering, 2025. URL https://www.nasa.gov/reference/2-0-fundamentals-of-systems-engineering/. Verification and validation distinctions and lifecycle constraints. Accessed 25 October 2026.

[25] National Center for O*NET Development. O*net online 13-1082.00: Project management specialists, 2026. URL https://www.onetonline.org/link/summary/13-1082.00. Occupational tasks; not time shares. Accessed 25 October 2026.

[26] National Center for O*NET Development. O*net online 15-1253.00: Software quality assurance analysts and testers, 2026. URL https://www.onetonline.org/link/summary/15-1253.00. Occupational tasks; not time shares. Accessed 25 October 2026.

[27] National Center for O*NET Development. O*net online 27-3042.00: Technical writers, 2026. URL https://www.onetonline.org/link/summary/27-3042.00. Occupational tasks; not time shares. Accessed 25 October 2026.

[28] National Institute of Standards and Technology. Digital twins for advanced manufacturing, 2026. URL https://www.nist.gov/programs-projects/digital-twins-advanced-manufacturing. Manufacturing interfaces, verification, validation and uncertainty quantification. Accessed 25 October 2026; standard work items are not claims of completed product certification.

[29] Shakked Noy and Whitney Zhang. Experimental evidence on the productivity effects of generative artificial intelligence, 2023. URL https://economics.mit.edu/sites/default/files/inline-files/Noy_Zhang_1.pdf. Working paper dated 2 March 2023, n=444; control mean 27 self-reported minutes, with 10–20 percent control ChatGPT use. Excluded as a strict no-AI benchmark. Do not conflate with the later 453-participant Science publication.

[30] OpenAI. Llm critics help catch llm bugs, 2024. URL https://cdn.openai.com/llm-critics-help-catch-llm-bugs-paper.pdf. Primary research paper; model criticism and human evaluation are distinct from the deterministic contract checks in this study.

[31] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. The impact of ai on developer productivity: Evidence from github copilot, 2023. URL https://arxiv.org/html/2302.06590v1. Version 1, 13 February 2023. Control mean 160.89 minutes conditional on completion; no Copilot, otherwise unconstrained; not a matched Alpha task.

[32] U.S. Bureau of Economic Analysis. Nipa handbook, chapter 10: Compensation of employees, 2023. URL https://www.bea.gov/resources/methodologies/nipa-handbook/pdf/chapter-10.pdf. December 2023. Domestic paid and national-income scopes distinguished; accessed 25 October 2026.

[33] U.S. Bureau of Economic Analysis. Gross domestic income: Compensation of employees, paid, GDICOMPA, 2026. URL https://fred.stlouisfed.org/series/GDICOMPA. Calendar 2025: USD 15,718.937 billion; annual, not seasonally adjusted; revision 30 September 2026; retrieved through Federal Reserve Bank of St. Louis, 25 October 2026.

[34] U.S. Bureau of Labor Statistics. Employer costs for employee compensation: December 2025, March 2026. URL https://www.bls.gov/news.release/archives/ecec_03202026.htm. Published 20 March 2026. Table 4: private industry occupations. Accessed 25 October 2026.

[35] U.S. Bureau of Labor Statistics. Table 1.2: Occupational projections, 2025–35, and worker characteristics, 2025, 2026. URL https://www.bls.gov/emp/tables/occupational-projections-and-characteristics.htm. 27 August 2026 edition; employment base year 2025, OEWS wages May 2025. Accessed 25 October 2026.