AGI ALPHA FLYWHEEL

The Advantage That Funds the Next

Useful Work → Productive Surplus
→ Greater Capability

SUCCESSOR Ω HARNESS

ACTIVATE. EVOLVE. SCALE.

Powered by TRANSFORMATIONAL INTELLIGENCE ∞
and the applicable $AGIALPHA utility.

Useful inventory checks.
Economic payback remains unproven.

A versioned executable asset, fresh public-source comparisons, strong conventional controls and financing bound to the exact tested capability.

25 October 2026, America/Toronto Edition 1.0
Local research execution. No realized revenue, production admission or self-acceleration is established.

Abstract

We construct a bounded release-inventory intake capability for the actual AGI ALPHA Workbench and its existing exact catalogue-comparison engine. Four public development inventories motivate the preparation bottleneck. Two same-session, model-assisted builders receive identical information and resource limits; both implement conventional parsers, and both freeze before independent confirmation. A strong ordinary interpreter retains standard recipes and the same input cache. The confirmation uses 128 pinned public source paths, partitioned into 16 distinct comparison requests across Alpha, Fleet, Jobs and Manager. Four representations per request are robustness repetitions, not additional jobs or customers.

All three methods produce correct complete results on all 16 requests and all 64 representations per method, correctly refuse 24 prohibited inputs, and pass eight secondary valid-boundary probes. The data contain two real source-identity changes. Median end-to-end pack times are approximately 23 milliseconds for every method. The experiment therefore delivers a useful preparation and discrepancy tool but does not establish a material incremental benefit over ordinary reusable software. Full attributable authoring, inference, operating and customer-attention costs remain unmetered.

The economic contribution binds each proposed investment to the exact executable asset tested, reuses Alpha's canonical cost and shadow-ledger contracts, and separates observed results from finite financing scenarios. Positive hypothetical attention savings are explicitly additional assumptions. They are not evidence that this candidate earned its development cost. The observed economic decision is HOLD. No real sale, transaction, token-derived operating surplus, distinctive builder superiority or sustained acceleration is claimed.

Principal finding. The retained asset works on fresh inputs. The strong conventional alternatives work equally well. Its full-cost payback has not been demonstrated; the financing mechanism must preserve that decision.

1 The question and the three claims

An economically useful Flywheel requires more than successful software execution. Production must yield a usable outcome; collected proceeds must exceed the complete costs and obligations; an affordable investment must improve later production enough to repay its construction and upkeep. These conditions concern different quantities and cannot be collapsed into a favorable unit-cost headline.

This study separates three claims. First, an automated service may have an economic advantage over equivalent competent human work without AI. Second, investing in and retaining a capability may improve the enterprise relative to continuing ordinary delivery. Third, Alpha's improvement method may outperform equally capable alternatives. A conventional component can support the first two claims. A tie against a conventional builder does not itself disprove investment value. However, value over a baseline that already retains the same practical capability cannot be inferred merely because the new package has a different name.

The user-facing objective is concrete: supply two permitted inventory documents, receive every declared source-identity discrepancy and a portable evidence pack, and understand whether further investment is warranted. The experiment makes

this tool available while keeping its economic claims bounded by the evidence.

2 Foundations and preserved findings

The current Alpha source is pinned to `95d1e97d`; canonical cost modules are copied without alteration from integration `e2503611`. The other three repositories contribute public source evidence and integration context; their deployed contracts and execution profiles are not modified.

Prior edition	Finding preserved in this edition
The Autonomous Work Economy	A retained quote specialist accepts 8/8 transfer cases; the conventional MILP also accepts 8/8, versus 2/8 for the initial capsule. This does not establish economical construction or distinct method superiority.
The Productive Capital Engine	Both retained and conventional caches reuse 114/132 cards. The child is about 0.9% slower on fresh change cycles; its modeled construction cost plus upkeep is not repaid.
The Economic Engine	All 144 primary financial paths buy zero improvements. A favorable higher-demand sensitivity stipulates capability performance rather than constructing the financed asset.
Build the Next Advantage	All five methods accept 96 unique jobs. Conventional active selection and the learned recipe each use 29 production labels; the learned recipe additionally consumes 31 training labels and remains on HOLD.

Table 1: Earlier editions remain unchanged. Their unfavorable findings are inputs to the new design, not results that this edition supersedes. Sources [3–6].

The new edition does not reactivate the held learned recipe. It retains ordinary parsing, explicit schema validation, exact hashing, normal caches and an existing comparison engine. Its new experimental connection is between a particular constructed asset, its fresh outcomes, its complete-cost uncertainty and the specific financing decision concerning that same asset. Original paper paths and publication requests remain separate and are not delayed for this study.

3 Bounded discovery before confirmation

The source inspection considered five opportunities. The choice was based on an observable preparation gap and plausible internal reuse, before the confirmatory inputs or outcomes existed. It was not selected because a favorable test result was already known.

Four actual Alpha inventories supply the development evidence: the free-try source map, manuscript source manifest, local-EVM source inventory and discoverability evidence manifest. They contain 8, 33, 55 and 45 records respectively. The ordinary workflow can retain mappings and scripts. We therefore do not assume that every later job repeats staffed preparation.

The declared economic horizons are 16, 52, 104 and 208 jobs. They are finite internal-workload scenarios, motivated by four maintained repositories; they are not measured release demand, customer orders or forecast sales. No market size, selling price or savings ratio is taken as established.

Opportunity	Prospective assessment
Inventory intake and packaging	Selected. Existing release evidence uses heterogeneous JSON objects, record arrays and text inventories. Alpha's comparison expects normalized CSV. Source preparation and strict interpretation are real integration work. Ordinary reusable recipes are a strong alternative.
Catalogue join speed	Rejected. One discovery run of the unchanged guide at 1,500 rows on each side took 123.717 ms. Absolute compute savings are too small to presume a credible finite dollar payback for development.
Batched Git reads and cache	Rejected as a primary opportunity. Standard batch interfaces and ordinary caches already exist. Withholding them would weaken the control artificially.
Planning-service reuse	Outside this study. Closed installed profiles and outstanding Harness, Fleet and security decisions remain.
Documentation and numerical examples	Rejected. The existing bounded domains have strong conventional operators and no identified material recurring cost.

Table 2: Discovery decisions, including rejected opportunities. The single timing is an exploratory measurement, not a performance estimate or demand study.

4 The productive asset

4.1 A narrow, executable contract

Each candidate exports `normalize(text, format='auto')`. It adopts standard parsing and explicit validation. It accepts one of four JSON inventory shapes, strict two-column CSV or SHA256SUMS. It returns exact, sorted path/SHA-256 pairs. An explicit relative `pathBase` is applied. Hexadecimal digest case is normalized; path case, whitespace, Unicode and percent-encoded characters are preserved exactly. It performs no filesystem or network access and executes no inspected source code.

Ambiguous roots, duplicate JSON keys, duplicate final paths, invalid hashes, unsupported hash types, traversal, absolute paths, malformed quoting and exhausted bounds must refuse. A Git SHA-1 object identifier is never substituted for a SHA-256 digest. These checks qualify the stated finite interpretation contract, not the safety or behavior of any program whose identity appears in the inventory.

4.2 Actual Alpha integration

The new research adapter calls the unchanged `guide.compute` implementation and runs through `createWorkbench(adapters)`. The Workbench supplies mission and source identities, bounded checkpoints, complete artifact digests, an inspectable pack and the existing inert Jobs lifecycle compilation. It continues to report local execution, economic admission HOLD, no authority and disconnected settlement. The inherited Jobs template language is generic rehearsal metadata, not a newly qualified source-assurance commission.

Delivered artifacts include a readable report, exact normalized inventories, complete comparison JSON, full differences CSV, proposed actions CSV and the retained asset identity. A saved pack can be inspected with the same Workbench. CSV downloads retain the guide's formula protection; exact original values remain in JSON and in the mission input. No discrepancy is automatically corrected.

The intake parser and the downstream service have different limits. The parser permits up to 1 MB of input, 128 records, depth 16 and 64 kB scalars. The unchanged TI guide imposes smaller normalized-CSV and cell bounds; the Workbench separately limits individual artifacts and canonical envelopes. A parser-valid input may therefore be refused by the complete service. The study does not broaden those limits or promise that every parser-valid document yields a Workbench pack.

4.3 Construction provenance

Two model-assisted construction agents receive the same public contract, development inputs, model capabilities, two-attempt limit and 20-minute elapsed allowance. Their output is reviewed local research code, not code automatically installed from a model reply. Construction happens in the current research session outside the installed Alpha operator runtime. We do not describe this as autonomous invention by a deployed enterprise.

The Alpha-side candidate uses one attempt and passes 62 development checks. Its recorded construction interval is 246.661 seconds. The conventional builder uses two attempts in 170 seconds. Its first version rejects a legal U+2028 path in SHA256SUMS; a development probe finds this and the second version corrects it, passing 39 checks. The initial source, defect and repair remain in its construction record. These are elapsed session intervals, not measured human labor or model bills. Tokens, inference cost and complete authoring cost remain unknown.

Both final sources freeze before confirmation generation. No confirmation-driven repair is applied. The prior learned methods remain inactive.

5 Confirmation design and controls

5.1 What constitutes one job

The collector takes the first 32 lexicographic eligible ordinary text paths from the union of each pinned current repository tree and its first-parent tree, then compares their bytes. NUL-containing, non-UTF-8 and individual files above 100 kB are excluded under a recorded prospective clarification. Four disjoint eight-file subsets per repository form 16 source-comparison jobs.

Each job is rendered in four representations: an observed native JSON shape, a whitespace/key-order JSON variation, CSV and SHA256SUMS. These are 64 evaluations per method and 192 method evaluations overall, but still 16 underlying comparison requests. The archived inputs are evaluator-derived inventories of real source bytes. They are not pre-existing release attestations or 16 independent release events. All raw bytes, Git identities and SHA-256 values are included for offline reproduction.

5.2 Three competent methods

The ordinary arm uses an independently assembled Python standard-library interpreter with saved format recipes. The Alpha and conventional retained arms use their separately constructed JavaScript modules. All three receive the same bounded

exact-input memoization policy. Successful prior preparations can be reused; unchanged old/new inventories need not be parsed twice.

The ordinary interpreter is new evaluation infrastructure. Its engineering and adoption are not claimed to be free or already installed in historical Alpha. It is a strong practical capability control, not a measured historical account of the enterprise's no-investment behavior. This limits the causal economic claim: the study cannot infer avoided historical interventions without observing them.

All methods use the same downstream Alpha comparison, packaging, acceptance and source evidence. Method order rotates across cases. The Python arm uses one persistent ephemeral child process; startup is reported separately, while IPC remains in end-to-end execution. Cross-language overhead is not evidence of a superior improvement algorithm.

5.3 Withholding and independent checking

Candidate hashes and the public protocol precede independent data generation. An evaluator that has not read either candidate implementation creates the confirmation sources and an independent Python oracle. Source-byte SHA-256 and length checks establish the expected normalized rows. A separate audit recomputes every produced pack, artifact and semantic result. Same-session role separation does not establish external institutional independence or a secret third-party holdout.

Twenty-four malformed or prohibited inputs test refusal. Eight secondary valid boundary cases test Unicode ordering, quoting and other exact accepted semantics. They are not counted as additional public-source jobs. The original protocol, two prospective clarifications, candidate freezes and generation receipts are preserved. Later accounting scenario details are labeled exploratory where they were chosen after technical outcomes were available.

6 Results on fresh work

Measure	Ordinary	Alpha asset	Conventional asset
Accepted unique requests	16/16	16/16	16/16
Accepted representation evaluations	64/64	64/64	64/64
Correct prohibited-input refusals	24/24	24/24	24/24
Valid secondary boundary cases	8/8	8/8	8/8
Incorrect accepted results	0	0	0
Median pack time (ms)	23.154	23.233	23.238
Median preparation time (ms)	0.552	0.232	0.275
Cache hits / misses, all probes	60/100	60/100	60/100

Table 3: One finite balanced-order confirmation run. The reported median is the upper middle order statistic (33rd of 64 values). Preparation is only a component of complete delivery; the ordinary arm includes Python IPC. No robust speed superiority is inferred from these small timing differences.

The ordinary method has the smallest median complete-pack time. Aggregate times over the 64 repeated representations differ in the other direction by only a few hundredths of a second. This sensitivity to the summary statistic reinforces the decision not to turn small timing differences into a material economic claim. The complete

local delivery pipeline includes ordinary Alpha planning, validation, hashing, comparison, packaging and pack inspection. It does not include customer intake, actual operator installation, commercial collection or all development work.

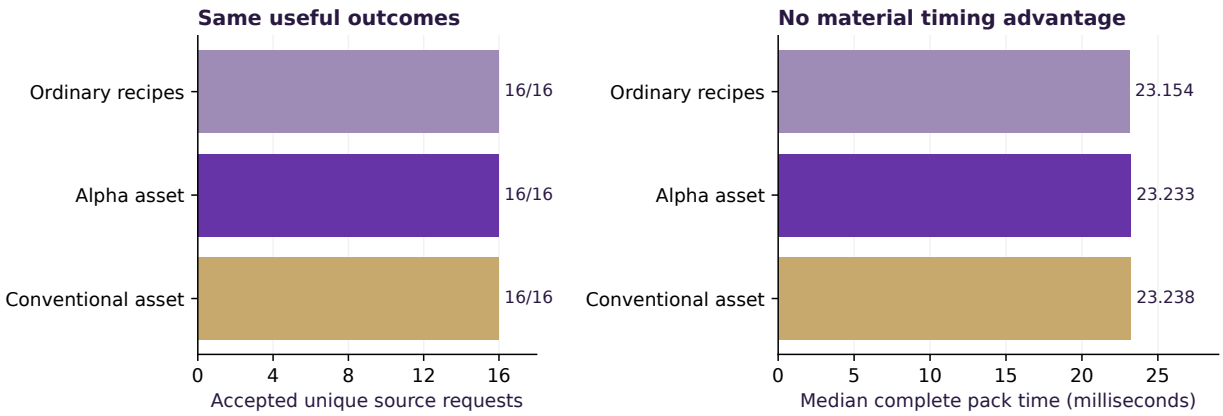


Figure 1: Fresh technical results. The full outcome ledger retains each evaluation; the three arms do not create three times as many unique jobs. The plot uses a zero time axis rather than magnifying small differences.

The independent Python audit performs 10,870 assertions across the 192 packs, including 1,152 artifact hashes, every complete comparison and CSV download, all source-byte identities and frozen dependencies. All assertions pass. This count measures repeated verification, not independent observations or customers. Four separately labeled post-primary diagnostics confirm downstream refusal at the guide/Workbench limits and successful delivery for empty inventories.

6.1 The useful result: two actual changed identities

Job J05 compares an eight-file Fleet subset between revisions `a4e2fec5` and `45759fb0`. The completed report identifies exactly two changed SHA-256 identities: `.github/workflows/local-skills.yml` and `README.md`. The other six selected files are unchanged. All other sampled subsets are unchanged. The demonstration uses this actual request, and readers can download the exact differences, actions and portable pack.

This is useful evidence for a bounded source-inventory task. A changed digest does not say whether a change is correct, safe or economically valuable. No Fleet code is executed and no held Fleet review is repeated. The service reports identities and leaves substantive source-review decisions with their existing owners.

7 Full-cost accounting without fabricated precision

7.1 Measured quantities and missing quantities

The execution ledger records each unique job, representation evaluation, method, output, source identity, failure, refusal and timing. Construction records retain the candidate versions, development checks, failed attempt and measured elapsed intervals. These observations do not provide a complete monetary invoice.

We reuse `alpha-automated-costs/v1` and `alpha-shadow-ledger/v1` unchanged. All eight oper-

ational stages are represented: preparation, planning, execution, verification, correction, packaging, reporting and recovery. Marginal, recurring and fully loaded costs remain distinct. Unknown quantities propagate into unknown totals. Zero task-time model calls do not make the authoring model, researcher, integration, maintenance or customer attention free.

For method a, complete horizon cost is

$$C_a = D_a + K_a + I_a + V_a + M_a + \sum_{j=1}^{N} \left(c_{aj}^{\text{run}} + c_{aj}^{\text{retry}} + c_{aj}^{\text{residual}} + c_{aj}^{\text{business}} \right).$$

Discovery D includes rejected opportunities. Construction K includes unsuccessful candidates. Integration I, qualification V, maintenance M, exceptions and failed delivery all remain costs. Common infrastructure must be charged or allocated symmetrically; its existence is not a zero-cost assumption.

The scenario investment envelope U is an explicitly assumed complete incremental budget for discovery, successful and failed construction, integration, qualification and unknown authoring/inference. It is not an observed charge. Maintenance and recurring delivery are charged separately. The grid spans 0, 5, 20, 50, 100, 250 and 500 USD; the zero value is only a boundary control. It cannot establish actual affordability.

7.2 Investment value and the price-cancellation test

At a common price P and accepted quantities Q_0, Q_1,

$$\Delta V_N = P(Q_1 - Q_0) - (C_1 - C_0).$$

When quality and accepted quantity are equal, the price term cancels. Raising an assumed selling price may make financing easier, but cannot turn a cost-dominated investment into an economic improvement. This is a central protection against manufactured economics.

For net recurring saving $s - m$ and complete fixed incremental cost F,

$$N^* = \left\lceil \frac{F}{s - m} \right\rceil, \qquad s > m.$$

Equality recovers cost; admission additionally requires strictly positive value. If $s \leq m$, there is no finite payback. Alpha's canonical investment calculator reports recovery at nonnegative net value; this study adds the strict-positive predicate required by the assignment. Affordability and technical acceptance are separate necessary conditions.

No recurring provider-attention saving was measured. Primary economic paths therefore assign no such benefit. A separate inverse frontier asks how many seconds of genuinely avoided work would be required. If w is an appropriate hourly cost and δc the per-job compute advantage,

$$t^* = \frac{3600}{w} \left(\frac{F}{N} + m - \delta c \right).$$

This is a threshold, not a claim that the asset actually avoids t^* seconds. The strongest ordinary recipes already deliver the same results automatically. Assigning them invented recurring manual labor would invalidate the comparison.

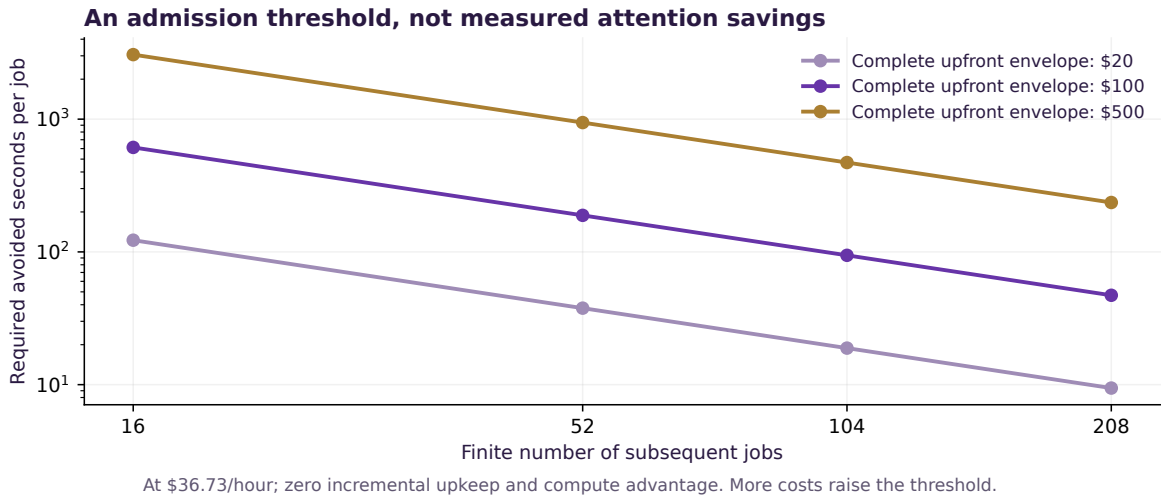


Figure 2: Inverse economic frontier. The assumed complete investment envelope is divided over a finite number of jobs. Additional upkeep or unallocated costs increase the required saving. No point is an observed attention benefit.

8 Financing the exact tested asset

The chronological model is a derived view over the existing canonical events, not another ledger. Each investment references the exact SHA-256 of the candidate that was constructed and tested. It never finances an unrelated hypothetical future capability while describing the tested candidate as successful.

The source-constrained sequence is: complete a mission, check acceptance, record delivery cost, wait for modeled collection, pay obligations, protect the next-job and maintenance reserve, inspect available earned contribution, and consider the identified investment. Only an affordable and positive-value candidate may enter the modeled deployment branch. Subsequent missions retain that same identity.

Initial cash is explicitly separate from earned operating contribution. A fixed 100 USD seed and 20 USD common setup expense are assumptions, not funds spent. Price, delay, lower demand, paid failures and maintenance shocks are varied. Scenarios stop at their finite horizon or resource exhaustion. No exhausted balance is replenished silently. Collections are simulated and never inserted as observed receipts in the canonical ledger.

The implementation exports 6,720 finite cash scenarios, 3,528 investment-frontier rows, 8,064 human-cost threshold rows and 1,680 sampled financing-price groups. Sixty representative scenarios retain every canonical event and projected cash transition. These are arithmetic scenarios, not additional executed missions. An independent Python audit verifies 126,396 arithmetic and conservation assertions.

All 3,360 primary cash scenarios make zero investments. The recorded median complete-delivery time provides no positive runtime advantage for either constructed asset, and no additional recurring attention saving was measured. The complete observed provider cost and investable surplus remain unknown.

An explicit secondary case illustrates the missing condition. Assume a 52-job horizon, a 20 USD complete incremental investment envelope, a 5 USD selling price, two-job collection delay, and an additional 60 seconds of provider attention saved on every retained use. The last quantity is **not observed**. Under these assumptions the model

funds the exact Alpha asset after job 13, retains it for 39 subsequent jobs, and first achieves strict cumulative payback at job 46. The terminal incremental benefit is 3.79 USD. With zero additional attention saving, the same investment is rejected. Adding 10 USD of declared maintenance exposure also prevents admission over the same horizon.

Across the secondary grid, 944 scenarios finance and repay the candidate under the extra benefit assumption. Across all stress scenarios, 364 stop when cash is depleted while preserving the specified reserve. No balance is replenished. The 5 USD-per-attempt cost-overrun stress was added during economic software development after the first exploratory run showed no exhaustion at the lower assumed cost; it is explicitly post-outcome analysis, not confirmatory evidence.

For perspective, a 20 USD complete investment with 0.01 USD incremental upkeep per use requires approximately 123.50, 38.68, 19.83 or 10.40 avoided seconds per use over 16, 52, 104 or 208 subsequent uses respectively, at the stated 36.73 USD/hour labor benchmark and 1 USD/hour compute assumption. These inverse thresholds do not establish that any such saving occurs. Waiting to accumulate collected contribution can require a longer total horizon than the simple post-investment calculation.

Positive secondary paths require an extra unobserved attention-saving assumption. They are labeled `COUNTERFACTUAL_BENEFIT_NOT_ESTABLISHED`. They demonstrate the arithmetic and identity binding of a possible financing mechanism; they do not demonstrate that this asset has earned its cost. The primary finding remains economic HOLD. A mechanism that refuses an unjustified purchase is operating correctly, even though it has not produced the desired growth.

9 Immediate human-work economics

Competent human work without AI may use ordinary spreadsheets, Git, scripts and verification tools. A cell-by-cell manual comparator would be a poor default. No recruited human panel or credible observed task duration is available here, so no measured percentage saving is asserted.

BLS May 2025 occupation data [8] report median wages of 65.38 USD/hour for software developers and 50.14 USD/hour for software quality-assurance analysts/testers. BLS June 2026 private-industry data [9] report 36.73 USD/hour in total compensation for office and administrative support occupations. These are sourced labor benchmarks, not billing rates, customer prices or task durations. An occupation's median wage is not silently converted into a measured loaded rate.

Provider and buyer thresholds answer different questions:

$$h^*_{\text{provider}} = \frac{60C_{\text{fully loaded}}}{w}, \qquad h^*_{\text{buyer}} = \frac{60(P + C_{\text{attention}} + C_{\text{switching}})}{w}.$$

At 36.73 USD/hour, each dollar of buyer total cost requires 1.634 minutes of equivalent displaced human work. Thus a 5 USD price with zero customer attention requires more than 8.168 minutes; a 10 USD price requires more than 16.335 minutes. These are arithmetic thresholds. If the competent worker could complete the task more quickly with ordinary tools, that comparison would not support the purchase at those prices.

Customer savings are not enterprise revenue. Provider cost, price, collected rev-

enue, reserves and investable surplus are reported separately. Unknown customer attention or switching cost makes the complete buyer comparison unknown.

10 Bilingual demonstration and reproduction

The offline English/French experience runs the actual frozen candidate and the actual Alpha Workbench comparison. One primary action opens the real J05 example. Users can inspect the raw inventories, select a supported format, run a bounded comparison and download complete artifacts. The result identifies the retained asset and makes the economic HOLD visible. A counterfactual financing panel labels its assumptions and does not present simulated collections as earned revenue.

The companion includes source, all archived inputs, independent expectations, outcome/cost ledgers, source hashes, failed development material, tests, plots, this manuscript and the standalone demonstration. Reproduction uses Node.js and Python's standard library; it does not call a model or require credentials, payments, live source retrieval or an installed operator profile.

```
python3 reproduce.py --output /absolute/new-directory
```

The command verifies the frozen inputs and dependencies, regenerates confirmation data from archived source bytes, executes the three arms, audits the resulting packs, recomputes the economic scenarios and checks the demonstration handlers. Timing values may vary; exact semantic outputs, source identities, refusal decisions and financial conservation must reproduce. Replaying code is not reproducing model construction or its unmetered inference cost.

The UI handler tests use a DOM shim; they are not browser rendering evidence. The prior browser restriction is preserved. Approved-operator, production and unattended persistent operation remain NOT RUN for this study. A downloadable working research demonstration is distinct from a newly deployed public service.

11 Token, execution and publication boundaries

The study makes no transaction. The Manager's USDC route and the applicable AGIALPHA route remain distinct. Existing source-inspected lifecycle and utility functions do not create new holder rights, distributions, buybacks or guaranteed returns. Token emissions, appreciation and customer savings contribute no spendable operating surplus in this accounting.

The existing source-assurance and cost owner, runtime/security/release owner and research publication owners keep their responsibilities under issue 4850. No original Harness/Fleet hold is retried, no security criteria are changed, and no new credential or persistent access is created. Local research correctness cannot clear the unrelated production security or release gates. An immutable repository source artifact is distinct from a served website publication.

This is an additive edition. It preserves all previous papers and results. The existing publication mechanism receives a reviewable source/PDF/demo package; this study does not create a competing publisher or override pending owner review.

12 What the evidence establishes

The actual constructed asset accepts fresh public-source inventories, preserves exact declared identities and produces useful complete discrepancy packs through Alpha. Independent checking finds no incorrect accepted result in this finite confirmation. Conventional reusable methods provide the same outcomes.

The experiment has not demonstrated the immediate human-work cost advantage, positive complete-cost incremental investment value, or a distinct Alpha builder advantage. It does not improve the builder itself, so improvement-of-improvement and sustained acceleration are not tested. The financing mechanism is concrete and source-bound; realized self-financing is not established.

The economic admission boundary is now inspectable. A further positive claim requires measured full costs and either genuine additional accepted work or genuine avoided production effort beyond strong ordinary reuse, at sufficient utilization. A higher assumed price, repeated formats or an unmetered authoring bill cannot supply that missing evidence. The useful tool can be retained as a research capability while its economic promotion remains held.

Claim	Decision from this study
Usable fresh deliverables	Supported within the finite intake and delivery contract.
Economic advantage over human work	Not established; sourced labor rates support thresholds, but equivalent duration and full costs are unknown.
Incremental retained-capability value	Not established over strong ordinary reuse. Primary modeled investment is zero.
Distinctive Alpha improvement method	Not established; the equally resourced conventional builder produces the same accepted outcomes.
Improved capability construction	Not tested on unseen construction problems; no acceleration claim.

Table 4: Separate decisions prevent a technical pass from becoming an unsupported economic or method claim.

After the primary economic HOLD, a separately recorded three-minute fallback screen examined reusable pack verification and builder fixtures. It took a recorded 116.522 seconds and inspected 13 source paths. Existing ordinary pack verification was already scripted and inexpensive in the recorded runs; ordinary development fixtures already retained the observed regression. Neither screen identified measured recurring burden, demand or subsequent construction savings that justified another framework. The decision was to retain those ordinary tools and construct no second speculative asset. This is post-primary discovery, not a second confirmation or evidence of growth; all original frozen files remain unchanged. The desired productive economic Flywheel remains unestablished.

References and source record

[1] AGI ALPHA, main source 95d1e97d4d1c0aaebd592767fc5e5d2da56f6aa7. https://github.com/MontrealAI/AGI-Alpha-Agent-v0

[2] Existing coordination and action boundaries, issue 4850. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/issues/4850

[3] The Autonomous Work Economy, original paper/companion lanes 4911/4912. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/pull/4911

[4] The Productive Capital Engine, original published PDF. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/blob/95d1e97d4d1c0aaebd592767fc5e5d2da56f6aa7/docs/papers/agi-alpha-productive-capital-v1.pdf

[5] The Economic Engine, original published PDF. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/blob/95d1e97d4d1c0aaebd592767fc5e5d2da56f6aa7/docs/papers/agi-alpha-economic-engine-v1.pdf

[6] Build the Next Advantage, original paper/companion lanes 4913/4914. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/pull/4913

[7] Canonical automatic-cost and shadow-ledger source, integration e2503611. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/tree/e2503611666ee170266cbce51dc250492285f2e8/examples/successor-alpha

[8] BLS, Occupational Employment and Wage Statistics, May 2025, Table 1. https://www.bls.gov/news.release/ocwage.t01.htm

[9] BLS, Employer Costs for Employee Compensation, June 2026, Table 4. https://www.bls.gov/news.release/ecec.t04.htm

[10] Standard comparison alternatives: pandas merge, join and compare documentation; Simon Willison's csv-diff. These are allowed practical alternatives, not evidence of matching this complete acceptance contract without integration. https://pandas.pydata.org/docs/user_guide/merging.html
https://github.com/simonw/csv-diff

Frozen identities

Alpha candidate:

`ad0b0adc9baa78de5b76cb7cf583fa8179fb977469ab5da47b75455953bf9fda`

Conventional candidate:

`ce44d675dc840b3bfd68e2bdb13d1aa5a5be230ff6a0318d430835a16826cb72`

Prospective protocol:

`f78f4a02556d48f79ceb116fe747c0d4515beb22c3df5b98a200b8ee60ce2df5`

Archived source record:

`7c0ccce536c6412c048581ce36d88f1cbebd3f48ef63677fa55ec393160ff349`

Declared adapter/method source binding in primary run:

`e437a024bd502bbc1c24052cf8d39bedeeea4ae7497186c99c0b308506850778`

This digest covers the adapter, normalizers and listed vendored dependencies. The runner, cache orchestration and confirmation data have separate commitments in the complete companion manifest.

Résumé français

Nous avons construit une capacité exécutable de préparation d'inventaires de livraison et l'avons reliée au véritable Workbench AGI ALPHA. Les trois méthodes testées produisent les mêmes résultats corrects sur 16 demandes distinctes et leurs 64 représentations par méthode. Elles refusent correctement les 24 entrées interdites et réussissent les huit cas limites valides. La démonstration révèle deux changements réels d'identité de fichiers, avec des rapports téléchargeables.

Cette réussite technique ne démontre pas encore un rendement économique positif. Les méthodes conventionnelles réutilisables obtiennent les mêmes résultats, le temps d'attention économisé n'a pas été mesuré et les coûts complets de création restent inconnus. Les scénarios de financement portent sur la capacité réellement construite et testée; leurs encaissements demeurent simulés. L'admission économique réelle reste en attente. Aucun revenu réalisé, aucune transaction, aucune supériorité propre de la méthode d'amélioration ni accélération soutenue n'est établi.