CLASSIFICATION PUBLIC **SUBJECT** PRODUCTIVITY MEASUREMENT **REVISION** 1.0

ISSUED SEPTEMBER 2026

The Hidden Factory

Manufacturing spent forty years learning that measuring the machine tells you nothing about the output. Enterprise AI is repeating the mistake, one token dashboard at a time.

A WHITE PAPER ON MEASURING HUMAN + AGENT PRODUCTIVITY

Every organization deploying AI agents at scale is measuring the wrong thing, and the reason is not carelessness. It is that the obvious measurement is the wrong one, and it takes an industry roughly a decade to discover this. Manufacturing already ran the experiment. In the 1980s, General Motors spent on the order of **$90 billion** automating its plants while continuing to measure productivity as direct labor hours per vehicle, a metric that mechanically improves whenever a human is replaced by a robot regardless of whether anything got better. Toyota, spending a fraction of that, measured yield, flow, and defect escape, and won.

The current generation of AI productivity metrics (tokens consumed, seats licensed, suggestion acceptance rate, percentage of code written by AI) are direct-labor-hours metrics in new clothing. They measure the utilization of the machine, not the throughput of the system, and several of them measure *cost* while being reported as *benefit*.

This paper does three things. It reconstructs what manufacturing actually learned and which instruments it built. It identifies six specific defects in how AI output is

measured today, each supported by published evidence. And it sets out a corrected framework with worked formulas, including a single derived inequality, `q > Î± + Ï`, that determines whether automating a given class of work creates or destroys value, and which explains why cheaper inference does not rescue a task whose output is expensive to verify.

---

01 THE PRECEDENT

What manufacturing learned the hard way

FOUR DECADES OF AUTOMATION, AND THE METRIC THAT BROKE FIRST

The industrial precedent is not a loose analogy. It is the same structural problem, a new class of machine absorbing work formerly done by people inside a measurement system built when only people did the work, and it produced a documented, expensive failure before it produced a solution.

1.1 The metric that inverted

Post-war manufacturing productivity was measured principally in **direct labor hours per unit**. It was a good metric for a good reason: labor was the dominant variable cost, and the number moved when the plant genuinely got better.

Robotics inverted it. Direct labor hours per unit falls when you remove a worker, whether or not the robot that replaced them produces a usable part. The metric could not distinguish between a plant that had become more productive and a plant that had merely moved its costs somewhere the metric did not look: into capital, into maintenance, into rework, into scrap. Worse, it actively rewarded the second behavior, because capital and rework were somebody else's line item.

1.2 GM: the $90 billion instrument error

Under CEO Roger Smith, General Motors pursued the "lights-out factory," a plant so automated it could run without illumination, because nobody would be in it. Roughly **$90 billion over a decade** went into automation and acquisitions. [1] The Detroit-Hamtramck plant became the emblem of what went wrong; the

widely-repeated story of robots painting each other rather than cars is probably embellished, but the substantive failures (robots installing windshields into the wrong position, systems that could not be recalibrated fast enough to keep the line moving) are not in dispute.[1][2]

The instructive part is what GM had already been shown. In 1984 it entered NUMMI, a joint venture with Toyota in a shuttered Fremont plant with a workforce GM had previously considered its worst. NUMMI reached Toyota-level quality with substantially *less* automation than GM's flagship plants. The lesson on offer was that Toyota's advantage was a system of work, not a class of equipment. As analyst Maryann Keller put it, Smith "thought Toyota possessed some kind of magicâ€¦ the trick to cost savings wasn't simply getting rid of people."[1]

MIT's Nelson Repenning and Don Kieffer have recently made this comparison explicitly as a warning about enterprise AI, noting that the same pattern (automating a process nobody has actually characterized) is producing the same result.[2]

1.3 Toyota: autonomation, not automation

Toyota's counter-principle, *jidoka*, is usually rendered as "automation with a human touch," and its defining feature is a machine that **stops itself** when it detects an abnormality rather than continuing to produce defects at speed. The point is not that the machine is slower. The point is that an unattended machine producing defective output faster is not a productivity gain; it is a defect generator with good throughput statistics.

This has a direct and uncomfortable translation to autonomous agents, which by default do exactly what the pre-*jidoka* machine did: continue confidently past the point where their output stopped being correct.

1.4 Tesla, 2018: the lesson does not stay learned

Three decades after Hamtramck, Tesla's Model 3 ramp stalled on an over-automated general assembly line. Elon Musk's public conclusion in April 2018 was unusually direct: *"Excessive automation at Tesla was a mistake. To be precise, my mistake. Humans are underrated."*[3] Tesla removed automation steps and reinstated human work to restore throughput.

The relevance is that this failure occurred at a company with no legacy process debt, excellent engineers, and full awareness of the GM precedent. The error is not a competence failure. It is what happens when a measurement system counts automated steps as progress.

1.5 The instruments manufacturing built

The resolution was not better robots. It was better instrumentation. Four ideas did most of the work, and all four have direct analogues that AI measurement currently lacks.

Instruments developed to measure automated production

AND THE QUESTION EACH WAS BUILT TO ANSWER

INSTRUMENT	WHAT IT MEASURES	THE FAILURE IT WAS BUILT TO CATCH
Overall Equipment Effectiveness Nakajima, TPM, 1988	Availability Ã— Performance Ã— Quality, as one multiplicative figure	A machine reported as "95% utilized" that was actually running slow and producing scrap. Multiplication makes any single weak factor impossible to hide.
First-Pass Yield / Rolled Throughput Yield	Share of units passing every step with no rework; the product of per-step yields	The *hidden factory*: rework capacity so normalized it never appears as a line item, because the unit eventually shipped.
Theory of Constraints Goldratt, 1984	System throughput as governed solely by the binding constraint	Local optimization: speeding up a non-bottleneck station, which raises work-in-process and lead time while throughput stays flat.
Little's Law L = Î»W	The fixed relation between WIP, arrival rate, and lead time	Believing you can raise arrival rate into a fixed-capacity stage without lead time or queue length responding.
Multifactor productivity	Output against *all* inputs: labor, capital, energy, materials	Labor-only metrics that improve automatically whenever cost shifts from labor to capital.

Note the common thread. Every one of these instruments exists to defeat a specific way of *looking* productive while not *being* productive. That is what a mature measurement system is: an accumulated set of defenses against the ways its predecessor could be fooled.

Six defects in how AI output is measured today

EACH ONE HAS A MANUFACTURING NAME AND A PUBLISHED COUNTER-EXAMPLE

The metrics currently reported in AI programs (to boards, to finance, and internally) cluster into what is best described as an **activity stack**: tokens consumed, API calls, monthly active seats, suggestion acceptance rate, percentage of code authored by AI, and self-reported time saved. Every item on that list is either an input, a proxy, or an opinion. None is an output.

DEFECT 01 · INVERSION

Cost is being reported as benefit

Token consumption is the money leaving the building. Presenting rising token volume as evidence of a successful AI program is structurally identical to a plant reporting rising electricity consumption as evidence of manufacturing excellence. It is a real number, correctly measured, pointing the wrong way.

The same applies to "percentage of code written by AI." That figure can rise because the tool got better or because engineers stopped reviewing carefully, and the metric cannot tell you which.

DEFECT 02 · THE HIDDEN FACTORY

There is no scrap rate, so rework is invisible

Manufacturing's most valuable discovery was that the rework loop is invisible unless you deliberately instrument it, the "hidden factory" of capacity consumed fixing things that were made wrong the first time. AI programs have no equivalent of first-pass yield, so an agent that produces an accepted-then-reverted artifact and one that produces a durable artifact are recorded identically.

EVIDENCE

GitClear's analysis of 211 million changed lines of code (2020 to 2024) found refactoring fall from roughly **25% of changed lines in 2021 to under 10% in 2024**, while copy-pasted code rose from **8.3% to 12.3%**, the first period on record in which copy/paste exceeded moved code. Duplication and declining refactoring are the signature of deferred rework, not of higher productivity.[4]

DEFECT 03 · THE VERIFICATION TAX

The cost of checking the work is not booked anywhere

This is the defect with no clean manufacturing precedent, and it is the most economically important one. In physical production, inspection is cheap relative to fabrication. You can gauge a machined part in seconds. For knowledge work, verifying that an artifact is correct can cost a large fraction of what producing it costs, and for some classes of work it costs *more*, because the reviewer must reconstruct the reasoning in order to trust the conclusion.

Every AI ROI model that counts inference cost and human time saved, but not human time spent verifying, is omitting what is usually the largest cost term. Section 4.5 shows that this single omission determines the sign of the answer.

EVIDENCE

In DORA's 2025 research, with AI adoption near **90%** of technology professionals, only **33%** of developers expressed trust in AI output and just **3%** reported high trust, while over **80%** believed AI increased their productivity.[5] Low trust is not a soft sentiment finding; it is a direct statement that output is being verified before use, and that verification is labor nobody is costing.

DEFECT 04 · LOCAL OPTIMIZATION

A faster stage is not a faster system

Goldratt's central result is that improving a non-constraint improves nothing. If AI accelerates code authoring while review, QA, and deployment approval capacity stay fixed, the result is not faster delivery. It is a larger queue in front of review, per Little's Law, and higher work-in-process.

This is not a theoretical concern; it is the observed pattern.

EVIDENCE

DORA's 2025 report characterizes AI's role as an **amplifier**: "it magnifies the strengths of high-performing organisations and the dysfunctions of struggling ones." It finds a positive relationship between AI adoption and delivery throughput *alongside continuing increases in delivery instability*, precisely the signature of a constraint being pushed harder rather than relieved.[5]

DEFECT 05 · INSTRUMENT BIAS

Self-reported time savings are not evidence

A large share of reported AI ROI rests on asking practitioners whether they felt faster. There is now a controlled measurement showing that this instrument does not merely have noise. It has sign error.

EVIDENCE

METR's randomized controlled trial (July 2025) had 16 experienced open-source developers complete 246 real tasks in mature repositories, with AI tools randomly permitted or forbidden per task. Developers expected AI to make them **24% faster**. They were measured **19% slower**. After completing the study, having personally experienced the slowdown, they still estimated AI had made them **20% faster**.[6]

The finding does not generalize to every context, and METR is careful about that: the participants were expert contributors in codebases they knew deeply, where the value of AI assistance is lowest. What generalizes is the **39-point gap between perceived and measured performance** in the same people on the same tasks. Any program relying on perception as its measurement instrument is exposed to an error of that magnitude, in either direction, and has no way to detect it.

DEFECT 06 · DENOMINATOR DRIFT

The task mix moves, and the metrics improve for the wrong reason

Teams route to AI the work AI handles well. This is correct behavior and bad measurement hygiene. As the mix shifts toward easier tasks, average cost per task falls, acceptance rate rises, and every headline metric improves, while the hard work that dominates the organization's actual cost base is untouched.

Without stratification by task class and weighting by baseline human effort, a portfolio metric will systematically flatter itself. Section 4.7 quantifies the distortion at roughly **1.6Ã—** for a plausible task mix.

2.7 What the aggregate evidence shows

These defects are not academic. MIT's NANDA initiative, reviewing 52 structured interviews, 153 senior-leader surveys, and over 300 publicly disclosed AI initiatives between January and June 2025, reported that roughly **95% of enterprise generative-AI pilots produced no measurable P&L impact**, against $30 to $40 billion of enterprise investment.[7]

The figure deserves care rather than repetition. It counts pilots that stalled before production, the six-month window is short, and the report itself notes ROI attribution is confounded. But a 95% null rate alongside near-universal adoption and 80%-plus perceived productivity gains is not primarily a technology finding. **It is a measurement finding.** Organizations cannot demonstrate value they have no instrument to detect, and they cannot stop value destruction they have no instrument to see.

There is a legitimate counter-reading, and it should be stated. Brynjolfsson, Rock and Syverson's *Productivity J-Curve* shows that general-purpose technologies reliably depress measured productivity early, because the complementary intangible investments (process redesign, new skills, reorganization) are costs now and returns later.[8] Solow's 1987 remark that "you can see the computer age everywhere but in the productivity statistics" preceded the late-1990s IT productivity surge by roughly a decade. Some of today's disappointing numbers are the downstroke of that curve.

That reading strengthens the argument rather than weakening it. You cannot distinguish "we are early on the J-curve" from "this is destroying value" without instruments capable of telling them apart, and the activity stack cannot. The framework below can: a J-curve organization shows rising first-pass yield against flat or negative net value, while a value-destroying one shows flat yield in the task classes that matter.

---

03 THE FRAMEWORK

A corrected measurement framework

SEVEN LAYERS, FROM EQUIPMENT EFFECTIVENESS TO PORTFOLIO ECONOMICS

What follows is a measurement stack, not a scorecard. Each layer answers a different question, and the layers are ordered so that each one's failure mode is caught by the next.

3.1 Layer 1 · Agent Operational Effectiveness

The direct transposition of Nakajima's OEE. Its virtue is multiplicative structure: three factors that each look acceptable produce a product that does not, which is exactly the disclosure property the activity stack lacks.

Agent Operational Effectiveness FORMULA 1 Â· AOE

AOE = A Ã— P Ã— Q

A

Availability. Share of scheduled operating time the agent could actually run. Discounts rate-limit rejections, tool and retrieval timeouts, orchestrator downtime, and context-window aborts.

P

Performance. (Attempts completed Ã— target cycle time) Ã· operating time. Penalizes context bloat, redundant tool calls, and recursive reasoning that advances no state.

Q

Quality, or first-pass yield. Attempts accepted downstream with no human rework, rejection, or schema fallback, over total attempts. This is the term the activity stack has no equivalent for.

Nakajima's world-class benchmark is `0.90 Ã— 0.95 Ã— 0.999 â‰ˆ 85%`. It is worth internalizing how far agent fleets sit from that: a fleet at `0.93 Ã— 0.70 Ã— 0.61` reports **39.7%** AOE, and each individual factor looks survivable in isolation. That is the disclosure property working.

3.2 Layer 2 · Rolled throughput yield across agent chains

Multi-step agent workflows are multi-station production lines, and yield multiplies rather than averages. This is the least intuitive and most consequential arithmetic in the framework.

Rolled Throughput Yield

FORMULA 2 Â· RTY

$$\mathbf{RTY} = â^{\prod}_{i=1..n}\ \mathrm{FPY}_i \qquad âŸ¹ \qquad \text{attempts per success} = 1 \,/\, \mathrm{RTY}$$

FPY_i

First-pass yield of step *i* in the chain: retrieval, planning, generation, tool execution, validation.

n

Number of steps that must all succeed for the work item to complete.

1/RTY

The cost multiplier on every input consumed. Failed attempts burn tokens, tool calls, and reviewer attention exactly as successful ones do.

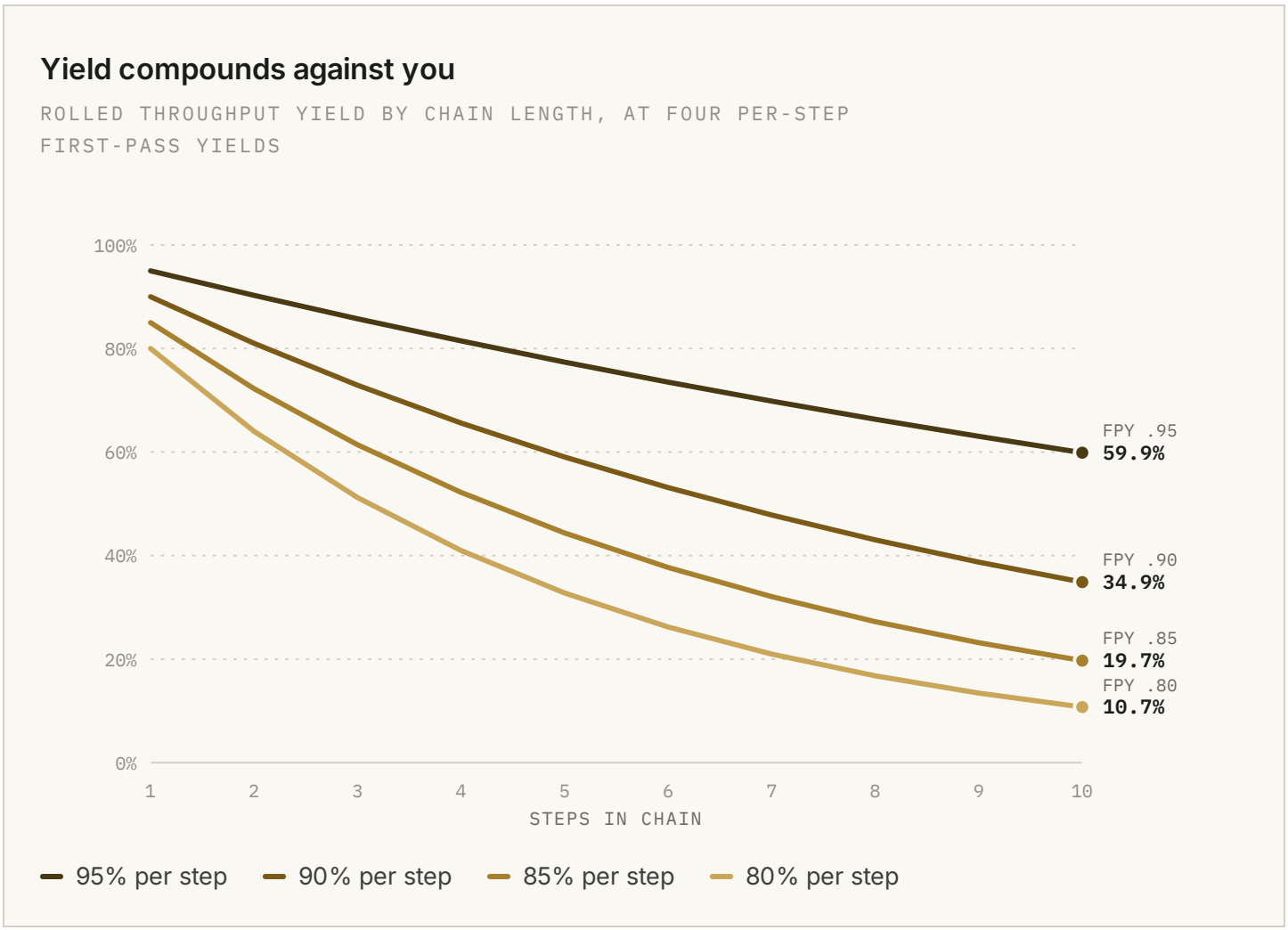


FIG. 1 A five-step agent chain with a highly respectable 90% yield at every step delivers a correct result **59% of the time** and costs 1.69Ã— its nominal price per successful outcome. At 80% per step, five steps yield 32.8% and cost 3.05Ã—. Per-step benchmarks reported in isolation are not merely incomplete. They are actively misleading about chain behavior.

3.3 Layer 3 · Marginal cost per successful task

The denominator is the entire argument. Cost per API call, cost per run, and cost per seat all divide by things the business does not buy. The business buys completed work.

Marginal Cost per Successful Task FORMULA 3 Â· MCST

$$\mathbf{MCST} = (C_{inf} + C_{tool} + \mathbf{C_{verify}} + \mathbf{C_{rework}}) / N_{success}$$

C_{inf}

Inference spend across **all attempts**, including every failed one. Failed runs are scrap and must be costed as scrap.

C_{tool}

Retrieval, sandbox, CI, and orchestration cost across all attempts.

C_{verify}

T_{review} Ã— $W_{reviewer}$ Ã— $N_{attempted}$. Charged against *every* attempt. You must inspect an output to learn whether it passed.

C_{rework}

T_{rework} Ã— W_{human} Ã— N_{failed}. The hidden factory, made visible.

$N_{success}$

Attempts accepted into a system of record without rework. Not attempts made; not artifacts produced.

3.4 Layer 4 · Net value added

MCST is a cost. Turning it into an economic verdict requires an honest counterfactual: what the same completed work would have cost without the agent.

Net Value Added & Productivity Ratio FORMULA 4 Â· NVA, Î

$$\mathbf{NVA} = (T_{baseline} \tilde{A}— W_{human}) \hat{a}^{\prime} MCST \quad \hat{A}\cdot \quad \mathbf{\hat{I}} = (T_{baseline} \tilde{A}— W_{human}$$

$T_{baseline}$

Measured hours for a competent human to complete this task class unaided. Must be measured, not assumed. It is the single most-fabricated input in AI ROI models.

W_{human}

Fully loaded hourly cost, from payroll or time-tracking systems rather than a flat per-head assumption.

Î

Productivity ratio. Î > 1 creates value; Î < 1 destroys it. Reporting Î by task class is the single highest-value chart an AI program can produce.

3.5 Layer 5 · The automation viability condition

Layers 1 through 4 measure a program already running. This layer decides, in advance, whether a class of work should be automated at all. It follows from three quantities and one line of algebra.

Let c_p be the cost of a human producing the artifact, c_a the cost of one agent attempt, and c_v the cost of a human verifying an agent-produced artifact to the same confidence. Normalize: Î± = c_a/c_p (relative inference cost) and Ï = c_v/c_p (**verification asymmetry**). Let q be first-pass yield.

Under a retry-until-success policy, expected cost per successful output is $(c_a + c_v)/q$, which beats c_p exactly when $(\hat{I}\pm + Ï)/q < 1$. Under a human-fallback-on-failure policy, expected cost is $c_a + c_v + (1\hat{a}^{\prime}q)c_p$, which beats c_p exactly when $\hat{I}\pm + Ï + 1\ \hat{a}^{\prime}\ q < 1$. Both reduce to the same inequality. The condition is invariant to failure policy, which is what makes it usable as a planning rule.

Automation Viability Condition

FORMULA 5 Â· THE DECISION GATE

$$q > \hat{I}\pm + \ddot{I}^{HOP} \quad \text{where} \quad \hat{I}\pm = c_a/c_p \ , \ \ddot{I}^{HOP} = c_v/c_p$$

q

First-pass yield for this task class.

Î±

Agent attempt cost as a fraction of human production cost. For current models on knowledge work this is typically 0.01 to 0.10 and falling.

Ï

Verification cost as a fraction of human production cost. Set by the *nature of the task*, not by the model, and it is not falling.

q*

Break-even yield, equal to Î± + Ï. Below it, automating this class destroys value at any volume.

THE CONSEQUENCE WORTH TAKING AWAY

Because Î± is small and shrinking while Ï is fixed by the task, the condition collapses in the limit to **q > Ï**: first-pass yield must exceed verification asymmetry. **Cheaper inference cannot rescue a task whose output is expensive to check.** A model 100Ã— cheaper moves Î± from 0.05 to 0.0005 and moves the break-even yield for a security-critical change from 0.85 to 0.80, against an actual yield of 0.60. The gap is not a pricing problem, and no amount of price movement closes it.

This also explains, without appeal to hype or incompetence, why AI adoption succeeds spectacularly on boilerplate and disappoints on judgment-dense work. Those are not different levels of model capability. They are different values of Ï.

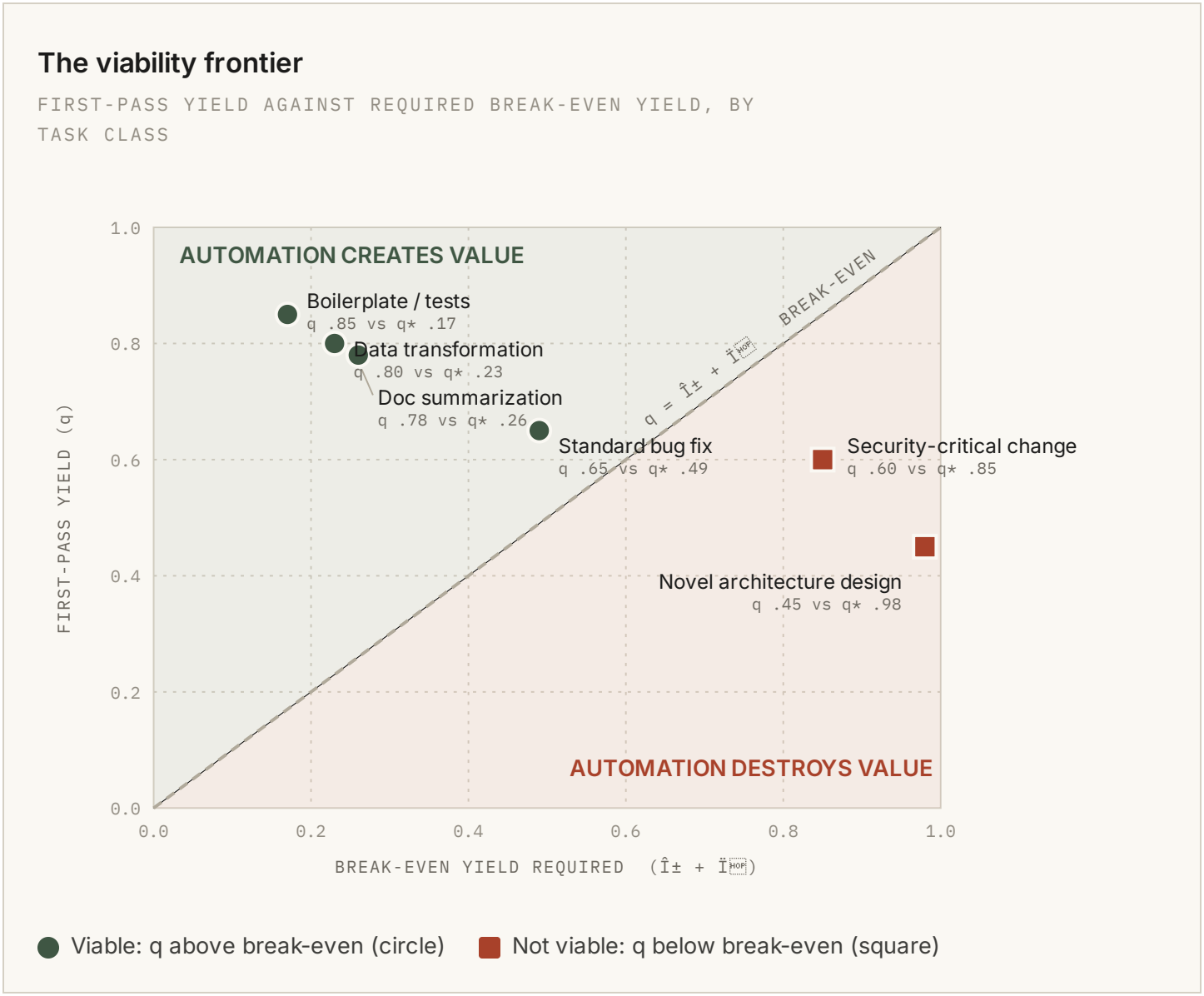


FIG. 2 The diagonal is the decision boundary. Distance from it, not distance from the origin, is what matters: "standard bug fix" at q = 0.65 is viable, while "security-critical change" at a barely-lower q = 0.60 is not, because verification for the latter costs 80% of what production costs. Two task classes with near-identical model performance land on opposite sides of the line. Illustrative values; each organization must measure its own ÏHOP per task class.

3.6 Layer 6 · System throughput

The preceding layers can all be satisfied while end-to-end delivery does not improve. Two classical results bound what a local speedup can do.

System Speedup & Queue Response FORMULA 6 Â· AMDAHL, LITTLE

$$S_{sys} = 1 / [(1 â^' f) + f/s] \quad Â· \quad L = Î» Ã— W$$

`f`
Share of end-to-end lead time occupied by the stage AI accelerates.

`s`
Local speedup achieved at that stage.

`L, Î», W`
Work-in-process, arrival rate, and lead time at the downstream constraint, typically human review.

The arithmetic is sobering. If authoring is 30% of lead time and AI makes it twice as fast, end-to-end improvement is `1/(0.70 + 0.15) = 1.176Ã—`, a **17.6% gain, not 100%**. If authoring became instantaneous, the ceiling is `1/0.70 = 1.43Ã—`, or **42.9%**. Everything beyond that requires relieving a different constraint.

Little's Law then supplies the failure mode. If review capacity absorbs 40 items per week and AI raises arrivals to 55, utilization exceeds 100% and the queue is unstable: work-in-process grows without bound and lead time diverges. Measured as "throughput up, instability up," which is exactly what DORA reports.[5] **The framework predicts the observed industry data.**

3.7 Layer 7 · Portfolio normalization

Finally, the guard against Defect 06. Portfolio metrics must be weighted by baseline human hours, not by attempt count, or the easy-task majority dominates the average.

Baseline-Hour-Weighted Portfolio AOE

FORMULA 7 Â· ANTI-GAMING

$$\mathrm{AOE}_{port} = \hat{I}\pounds_k\ w_k\ \hat{A}\cdot\ \mathrm{AOE}_k \quad \text{where} \quad w_k = n_k T_{base,k} \,/\, \hat{I}\pounds_j\ n_j T_{base,j}$$

k

Task class, stratified by complexity, not by team, tool, or department.

w_k

Weight by share of *baseline human hours* the class represents, so a class that consumed 55% of the organization's effort carries 55% of the metric.

Attempt-weighted versus hour-weighted portfolio AOE

ILLUSTRATIVE MIX, SAME UNDERLYING PERFORMANCE, TWO WAYS OF AVERAGING

TASK CLASS	ATTEMPT SHARE	BASELINE HRS	AOE	HOUR WEIGHT
Trivial: boilerplate	60%	0.3	86%	12.3%
Moderate: bug fix	30%	1.6	55%	32.9%
Complex: design	10%	8.0	28%	54.8%
Weighted by attempt count FLATTERING			**n/a**	**70.9%**
Weighted by baseline hours HONEST			**n/a**	**44.0%**

The unweighted figure overstates performance by **26.9 percentage points, a factor of 1.61Ã—**. No individual number in the table is wrong. Only the averaging is.

04 WORKED CASE

One task class, one quarter, fully costed

WHERE THE MONEY ACTUALLY IS

The following applies the framework to a single task class, "standard bug fix," over one quarter in an organization running roughly 12,000 agent attempts against it. Every input is a quantity an organization can measure from systems it already operates.

Inputs

ALL VALUES MEASURABLE FROM EXISTING SYSTEMS OF RECORD

QUANTITY	VALUE	SOURCE SYSTEM
Attempts, $N_{attempted}$	12,000	Gateway / orchestration logs
First-pass yield, q	62%	Issue tracker: resolved without reopen or rework tag
Inference cost per attempt	$0.38	Provider billing API
Tooling cost per attempt	$0.06	CI / sandbox / retrieval metering
Review time per attempt	0.20 hr	Time-tracking or PR review telemetry
Rework time per failed attempt	0.75 hr	Time-tracking, reopened-issue duration
Reviewer loaded rate, $W_{reviewer}$	$95/hr	HRIS / payroll
Engineer loaded rate, W_{human}	$85/hr	HRIS / payroll
Unaided baseline, $T_{baseline}$	1.60 hr	Measured control group, *not assumed*

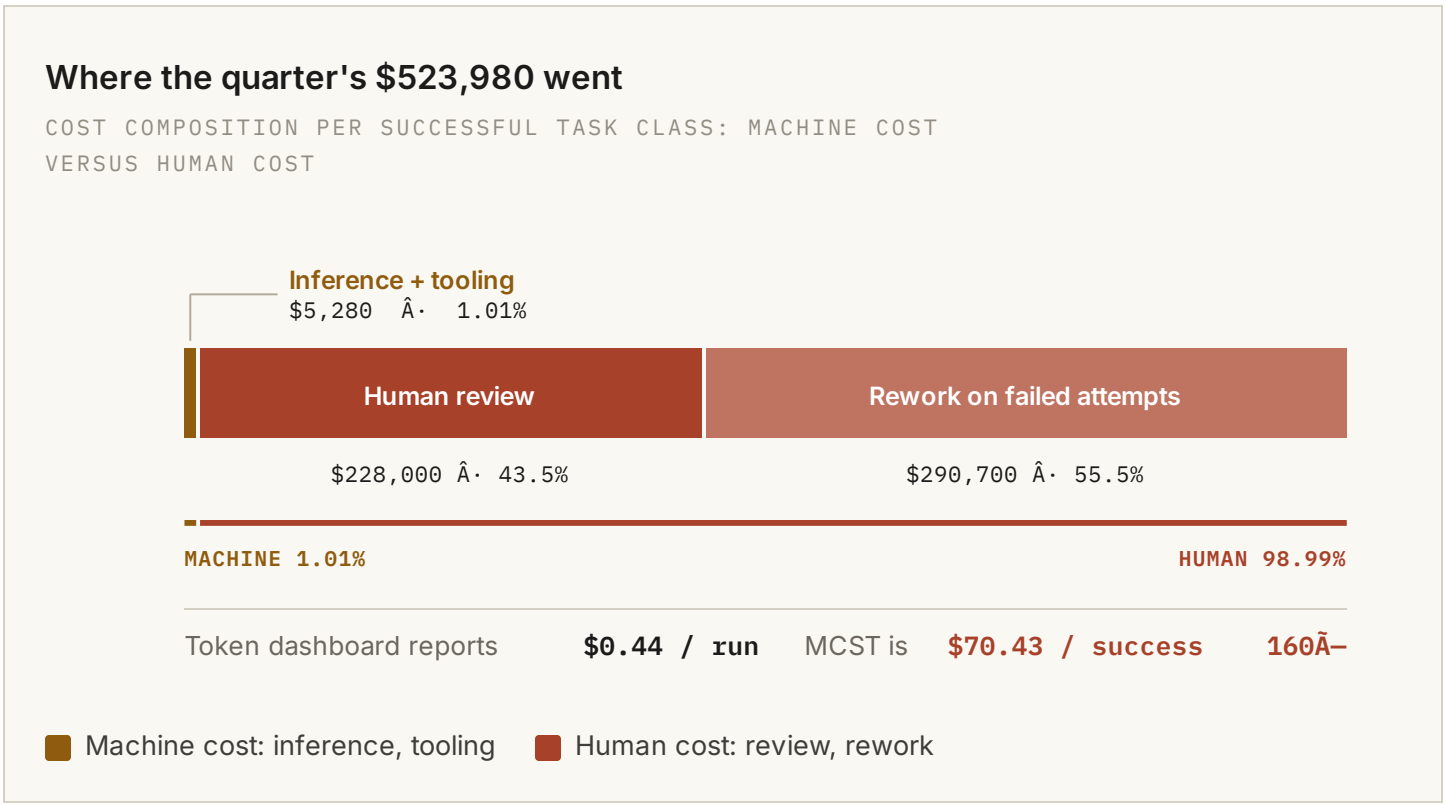


FIG. 3 Inference and tooling together are **1.01%** of the true cost of this task class. The cost-per-run figure a token dashboard displays ($0.44) understates the marginal cost of a successful outcome by a factor of **160**. An organization optimizing the visible 1% while the invisible 99% goes unmeasured is not running a cost program; it is running a rounding error.

Quarterly result

12,000 ATTEMPTS Â· 7,440 SUCCESSES Â· STANDARD BUG FIX CLASS

LINE	AMOUNT	SHARE
Inference, all attempts	$4,560	0.9%
Tooling and infrastructure	$720	0.1%
Human review, all attempts	$228,000	43.5%
Rework, 4,560 failed attempts	$290,700	55.5%
Total cost	**$523,980**	**100%**
MCST, cost per successful task	$70.43	n/a
Human baseline value, 1.6 hr Ã— $85	$136.00	n/a
Net value added per successful task	$65.57	n/a
Productivity ratio Î	1.93	VALUE CREATING
Quarterly net value added	**$487,860**	**n/a**

This program is genuinely working: Î = 1.93 means the class returns roughly $1.93 of avoided human cost per $1.00 spent. The viability condition agrees independently. Here Î± = 0.003 and Ï = 0.140, so q* = 0.143 against an actual q of 0.62, comfortably clear. **The two methods must agree in sign, and do; that consistency is the framework's internal check.**

4.1 The sensitivity that should reset the roadmap

Having built the model, the useful question is which lever moves it.

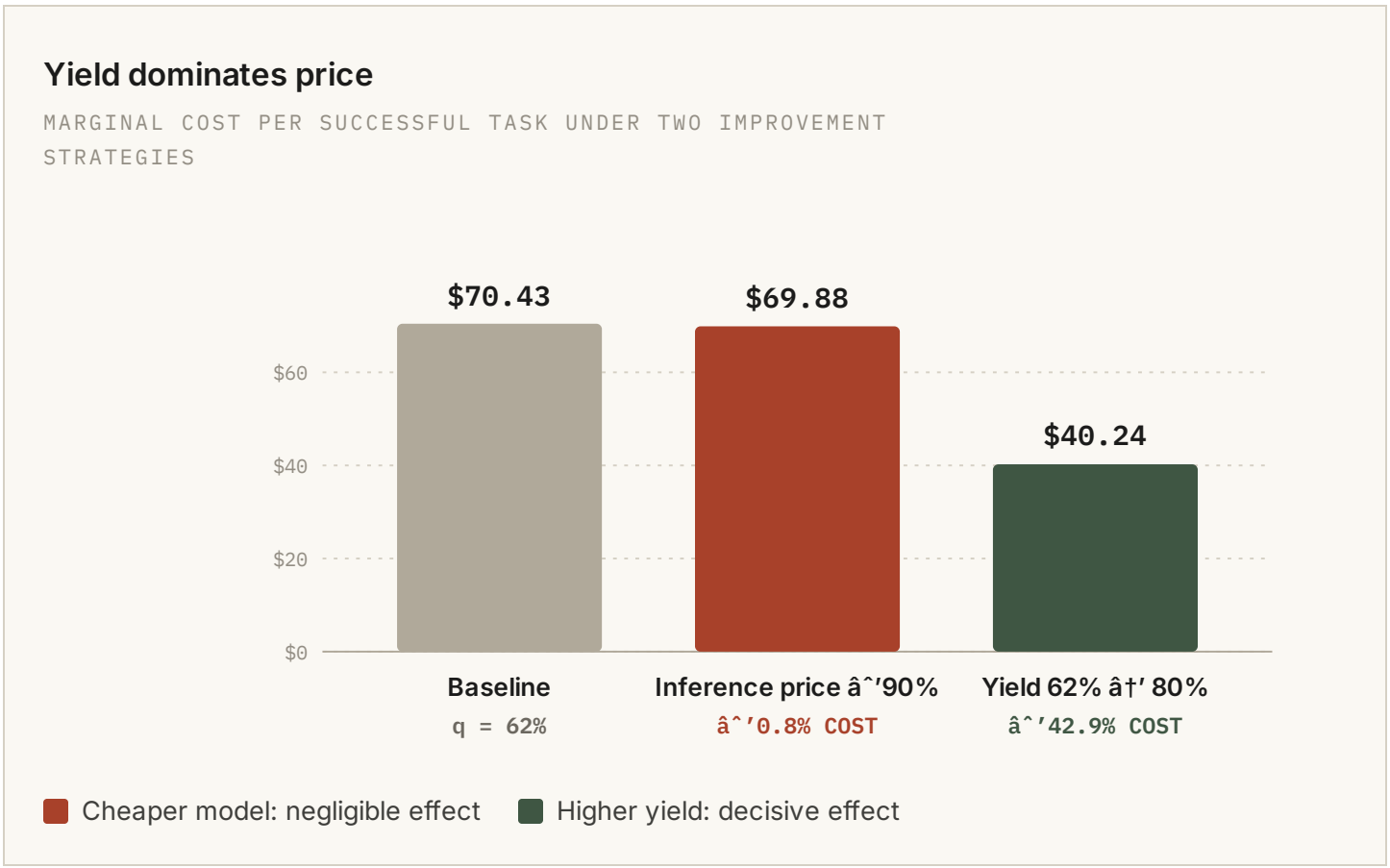


FIG. 4 Cutting inference price by 90% reduces cost per successful task by **0.8%**. Raising first-pass yield from 62% to 80% reduces it by **42.9%** and nearly doubles quarterly net value added, from $487,860 to $919,320. Model-shopping and prompt-caching optimize a term worth one percent of the total; evaluation, scoping and routing optimize the term worth ninety-nine.

This is the practical payoff of measuring correctly. An organization whose dashboard shows only token spend will spend its optimization effort on the 1% line, because it is the only line it can see. The framework redirects that effort to yield, where it is worth roughly fifty times more.

---

05 ADOPTION

Instrumentation and maturity

WHAT TO COLLECT, AND THE ORDER TO EARN IT IN

The framework requires no payload inspection, no proxy in the request path, and no new agent behavior. Every input is metadata already emitted by systems the organization runs.

Minimum instrumentation

METADATA ONLY, NO PROMPT OR PAYLOAD CAPTURE REQUIRED

SIGNAL	SOURCE	FEEDS
Tokens, cost, latency, status codes per attempt	Provider billing APIs, gateway logs	C_{inf}, Î±, Availability
Attempt â†’ work-item correlation ID	Orchestrator, injected trace header	Every join in the framework
Work item resolved / reopened / reverted	Issue tracker, VCS, CRM, ticketing	q, RTY, $N_{success}$
Review duration and outcome	PR review telemetry, time tracking	C_{verify}, Î
Rework duration on failed attempts	Reopened-issue lifecycle, time tracking	C_{rework}
Loaded hourly cost by role	HRIS / payroll	W_{human}, $W_{reviewer}$
Unaided completion time by task class	Held-out control group or pre-adoption history	$T_{baseline}$ HARDEST

Two of these deserve emphasis because they are where programs fail. The **correlation ID** is the entire framework's load-bearing element: without a durable join between a model call and the business object it advanced, none of the formulas can be computed and the organization is permanently stuck at activity metrics. It costs almost nothing to add at the start and is expensive to retrofit.

The **baseline** is the one input that cannot be harvested from logs. It requires either pre-adoption history or a deliberately maintained control group. Most organizations substitute an assumption here, and because NVA is linear in $T_{baseline}$, that assumption silently determines the answer. A program that assumes its baseline has not measured its ROI; it has restated its own hypothesis in currency.

5.1 Maturity ladder

L0 **Activity**

Tokens, seats, API calls, acceptance rate, self-reported time saved. No connection between spend and outcome. Cannot answer whether the program creates value.

WHERE MOST ORGANIZATIONS ARE TODAY

L1 **Attribution**

Every model call joined to the work item it advanced, and to a team and cost center. Cost per completed work item becomes computable for the first time.

UNLOCKS Â· MCST DENOMINATOR

L2 **Yield**

First-pass yield, rework, and reversion tracked per task class. Failed attempts costed as scrap. The hidden factory becomes a line item.

UNLOCKS Â· AOE, RTY, FORMULAS 1 AND 2

L3 **Economics**

Verification and rework labor costed against real wage data; measured baselines by task class. Net value added and Î reportable per class, defensible to finance.

UNLOCKS Â· MCST, NVA, Ï, FORMULAS 3 TO 5

L4 **Control**

Routing decided by measured q and Ï rather than by model benchmark. Work whose q falls below q* is flagged for return to people, and the decision is made and confirmed by a named owner. Capacity and spend forecast from historical variance; constraint analysis before any local speedup is pursued.

TARGET STATE Â· DECISIONS STAY WITH PEOPLE Â· AVYV MEASURES AND RECOMMENDS, IT DOES NOT ROUTE OR ENFORCE

The ladder is strictly ordered. Yield cannot be measured before attribution exists, and economics cannot be computed before yield. Programs that attempt to report board-grade ROI from L0 data are not measuring; they are estimating, and Defect 05 governs the accuracy of estimates.

06 LIMITS

Objections and honest limits

WHERE THIS FRAMEWORK IS WEAKEST

GOODHART'S LAW APPLIES TO THIS FRAMEWORK TOO

First-pass yield can be gamed by loosening acceptance criteria, and MCST by routing only easy work. Formula 7 defends against the second; nothing defends against the first except keeping acceptance definitions owned by the function that consumes the output, never by the function whose performance is being measured. This is the same governance separation that keeps a quality department independent of production in manufacturing, and it is a governance answer, not a technical one.

MEASUREMENT IS NOT FREE

Review-time capture is the expensive input, because it is the one least likely to be already instrumented. A reasonable compromise is periodic sampling of review duration by task class rather than continuous capture: Ï is a structural property of a task class and moves slowly, so a quarterly sample is usually sufficient. Continuous capture is warranted only where a class sits near the viability boundary.

THE VERIFICATION ASYMMETRY Ï IS GENUINELY HARD TO MEASURE

It requires knowing what verification to the same confidence costs, and "the same confidence" is doing real work in that sentence. In practice organizations will estimate Ï from review-time telemetry and refine it from escaped-defect rates. This is the framework's softest input and should be reported as a range, with the viability verdict tested across it.

TASK HETEROGENEITY RESISTS STRATIFICATION

Knowledge work does not partition as cleanly as machined parts. Stratification will be imperfect and will need periodic revision as the work itself changes shape under automation. Imperfect stratification is nonetheless strictly better than none, which is the status quo.

THE J-CURVE CONFOUND IS REAL

Early negative NVA may reflect complementary investment not yet paying off rather than genuine value destruction.[8] The distinguishing signal is the *trajectory of q*, not the level of NVA: a J-curve organization shows first-pass yield improving quarter over quarter while NVA is still negative. A genuinely failing deployment shows flat q. Reporting q as a time series, not a snapshot, is what makes the two separable, and no activity-stack metric can make that distinction at all.

NOT EVERYTHING VALUABLE IS A TASK

Exploratory work, learning, and the option value of capability that has not yet been applied are real and are not captured by a completed-task denominator. The framework measures production, and should not be extended to govern research. Applying task economics to exploratory work would reproduce, in a new domain, precisely the error this paper describes.

---

07 CONCLUSION

Measure the output, not the machine

The argument reduces to one sentence: **an organization cannot manage what its instruments cannot distinguish**, and the current instruments cannot distinguish a productive AI deployment from an expensive one.

Manufacturing reached this conclusion between roughly 1985 and 1995, at a cost measured in tens of billions of dollars and several near-death corporate experiences. It responded by building instruments (OEE, first-pass yield, rolled throughput yield, constraint analysis) whose common property is that each defeats a specific way of appearing productive without being productive. Those instruments transfer. The transposition is not metaphorical; the mathematics is the same mathematics, because the structure of the problem is the same structure.

Three claims are worth carrying out of this paper:

- **Yield compounds, and chains are worse than their steps.** Five 90% steps produce a 59% result at 1.69Ã— the nominal cost. Per-step benchmarks are not merely optimistic about chain behavior; they are the wrong quantity.
- **Verification cost, not inference cost, decides viability.** The condition `q > Î± + Ï` holds regardless of failure policy, and because Î± is small and

falling while Ĩ is fixed by the nature of the task, cheaper models do not change the answer for judgment-dense work. In the worked case, a 90% price cut moved unit economics by 0.8%; an 18-point yield improvement moved them by 42.9%.

- **The measurement gap, not the capability gap, is what most organizations are actually facing.** A 95% null-ROI rate coexisting with 80%-plus perceived productivity gains is the signature of a missing instrument, not a missing technology.

The organizations that will defend their AI spend to a board in two years are not the ones with the largest deployments or the cheapest tokens. They are the ones that can produce a number, per class of work, showing what a completed unit of output cost and what it would have cost otherwise, and show the raw records underneath it. That is a measurement capability, and it is built before it is needed, not after it is demanded.

Toyota did not win because it had better robots. It won because it knew, station by station, what it was actually producing.

---

08 REFERENCES

References

[1] Graban, M. *"GM's CEO Roger Smith Thought Toyota Had Magic, But This Was the Secret."* Lean Blog, June 2016. On GM's ~$90bn automation program, the Hamtramck failures, the NUMMI joint venture, and Maryann Keller's assessment. leanblog.org (https://www.leanblog.org/2016/06/gms-ceo-roger-smith-thought-toyota-had-magic-but-this-was-the-secret/)

[2] Repenning, N. and Kieffer, D., MIT Sloan School of Management, reported in *Fortune*, 3 September 2025. GM versus Toyota in the 1980s as a case study for enterprise AI adoption. fortune.com (https://fortune.com/2025/09/03/case-study-mit-general-motors-toyota-1980s-artificial-intelligence-lesson/)

[3] Musk, E., April 2018, reported in *TechCrunch*. "Excessive automation at Tesla was a mistakeâ€¦ humans are underrated." techcrunch.com (https://techcrunch.com/2018/04/13/elon-musk-says-humans-are-underrated-calls-teslas-excessive-automation-a-mistake/)

[4] GitClear. *AI Copilot Code Quality: 2025 Research.* 211 million changed lines, 2020 to 2024; refactoring 25%â†'<10%, copy-

paste 8.3%â†’12.3%. gitclear.com (https://www.gitclear.com/ai_assistant_code_quality_2025_research)

[5] DORA. *State of AI-assisted Software Development, 2025*, as reported by InfoQ. ~90% adoption; AI as "amplifier"; throughput up alongside rising instability; 33% trust, 3% high trust, 80%+ perceived productivity gain. infoq.com (https://www.infoq.com/news/2025/09/dora-state-of-ai-in-dev-2025)

[6] METR. *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity.* July 2025. 16 developers, 246 tasks; expected +24%, measured â^’19%, believed +20% afterwards. metr.org (https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/) Â· arXiv:2507.09089 (https://arxiv.org/abs/2507.09089)

[7] MIT NANDA. *The GenAI Divide: State of AI in Business 2025.* 52 interviews, 153 leader surveys, 300+ disclosed initiatives, January to June 2025; ~95% of pilots without measurable P&L impact. State of AI in Business 2025 (PDF) (https://mlq.ai/media/quarterly_decks/v0.1_State_of_AI_in_Business_2025_Report.pdf)

[8] Brynjolfsson, E., Rock, D. and Syverson, C. *"The Productivity J-Curve: How Intangibles Complement General Purpose Technologies."* American Economic Journal: Macroeconomics 13(1), 2021, pages 333 to 372. NBER w25148 (https://www.nber.org/papers/w25148)

[9] Nakajima, S. *Introduction to TPM: Total Productive Maintenance.* Productivity Press, 1988. Origin of Overall Equipment Effectiveness and the 85% world-class benchmark.

[10] Goldratt, E. M. and Cox, J. *The Goal: A Process of Ongoing Improvement.* North River Press, 1984. Theory of Constraints.

[11] Womack, J. P., Jones, D. T. and Roos, D. *The Machine That Changed the World.* Rawson Associates, 1990. The MIT International Motor Vehicle Program's comparison of mass and lean production.

[12] Solow, R. *New York Times Book Review*, 12 July 1987. "You can see the computer age everywhere but in the productivity statistics."

FIGURES 1 TO 4 COMPUTED FROM THE FORMULAS IN Â§3; ILLUSTRATIVE INPUTS STATED IN Â§4