Journal of Business and Management Studies
ISSN: 2709-0876
DOI: 10.32996/jbms
Journal Homepage: al-kindipublisher.com/index.php/jbms/index

From cost-of-pass to cost per successful task: A task-level accounting protocol for agentic AI products

Krunal Panchal ✉[1], **Nixalkumar Patel**[2]

[1] *Independent Scholar, United States*

[2] *Independent Researcher, New York, United States*

Corresponding Author: Krunal Panchal**, E-mail:** Krunal.cp@live.com

ABSTRACT

An agentic AI product can make multiple calls to models, use external tools, try things a few times, verify a lot of things, call for human input, etc. to complete a single user request. These are pieces of this process, but do not, individually, represent the cost of a done job. Previous work has shown that cost-of-pass is the anticipated cost to finding a correct solution, and recent studies have begun to associate cost with the outcomes that a successful solution would yield. This paper does not therefore suggest a new economic concept, cost per successful task (CPST). Instead, it defines an accounting protocol at the task level for using outcome based costing for bounded agentic workflows. The protocol specifies a task and its definition of success before execution, assigns variable costs to each task started and unsuccessful tasks to the cost numerator and reports both the cost and the success, latency, human intervention and policy compliance. It splits up the costs of execution and costs of allocated platform and downstream losses. We derive the relationship between CPST and success probability, describe the relationship between bounded retries and that same relationship, and propose an evaluation design to test whether or not the bounded retries change the product decisions based on the cost of that product vs. its model price. The contribution is a specification for a measure, not an empirical measure, and a set of propositions which are testable, not empirical claims.

KEYWORDS

agentic AI; AI unit economics; cost-of-pass; task success; workflow evaluation; FinOps; human intervention

ARTICLE INFORMATION

ACCEPTED: 30 September 2026 **PUBLISHED:** 31 October 2026 **DOI:** 10.32996/jbms.2026.8.11.5

1. Introduction

The economic unit to measure an AI product should match the product's expected outcome. An enterprise workflow can need to retrieve, execute the tool, verify, approve, and retrieve again before the user's goal is achieved, or a model call might return an answer. FinOps guidance is able to differentiate between resource measures, like cost per token, and business measures like cost per transaction or resolved case. It also suggests linking generative AI investments to units with goals.

The basic principle of allocating spending according to results is set out. The principle of allocating expenditure on the basis of successful outcomes is put in place. Erol et al. give a definition of cost-of-pass based on the expected cost of an inference attempt and its probability of yielding the correct solution. Wang and colleagues subsequently use cost-of-pass to study agent architectures. In this case, it would be pointless to rename this quantity to CPST and claim it as a research contribution.

One problem yet exists with the unit being a bounded workflow instead of a single inference attempt. Product teams need to determine when a task starts and ends, what is considered successful, what tools and human effort should be assigned to a particular task, how to count task failures and abandoned tasks, and which costs are included in a comparable measure of operation. This paper is meant to set forth explicit rules for those decisions. It is a protocol that can be reused, tested and revised—it's not a discovery of outcome-based costing.

Copyright: © 2026 the Author(s). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) 4.0 license (https://creativecommons.org/licenses/by/4.0/). Published by Al-Kindi Centre for Research and Development, London, United Kingdom.

2. Related Work and Scope of Contribution

The protocol is based on three sets of work.

First, there is a cost of pass joins, as well as correctness. With the independent-trial assumptions stated, Erol et al. give the expected cost of a correct solution as expected cost per attempt divided by the probability of correctness. They also study economic borders and human-expert bases. Has the tightest formal tie to CPST.

Secondly, agent benchmarks demonstrate the need to define the outcome at the task level. In τ-bench, the accuracy of an agent's behavior is measured against a target database state after interacting with a simulated user and tools; repeated tests measure behavioral consistency. SWE-bench tests if a proposed change to the code addresses an issue in an execution environment. Both of these are different ways of defining success but they both refer to tasks that have been completed.

Third, the FinOps practice already has the concept of business outcome units. It's guidance comprises cost per resolved case, and it asks for documented cost envelopes and sources. A more recent guidance for using AI terms use case economics as the cost of realizing a desired outcome. The question here discussed is not whether unit economics should exist, but how can an agentic task be measured in a consistent manner consistently?

3. Task and Success Specification

3.1 Unit of observation

A task is one request that is initiated to deliver a predetermined user or business goal within a certain workflow policy. The task record starts from the moment that the system accepts the request. It terminates when the decision is accepted, rejected, abandoned, or when the decision is escalated beyond the defined workflow, or when a prespecified time limit is reached, or when a prespecified resource limit is reached.

Multiple model calls, tool calls, retries and human actions can be present within a task. The following steps have the same task identifier. If a task needs to be retried during the workflow, the same task will be retried, not a new one. When a customer sends in a new request after a previous request has been closed with the workflow, the new request is given a new identifier, the difference between the two must be defined prior to measurement.

In comparison, the tasks should be aggregated by stable task description and population of workload. A simple information lookup as well as a complex incident resolution should not be combined for no more than the fact that both are referred to as "tasks."

3.2 Successful completion

Prior to evaluation, the product owner and/or domain experts should define an observable success rule. For task i, let:

$$S_i = \begin{cases} 1, & \text{if the required outcome is achieved, accepted, and within the stated constraints;} \\ 0, & \text{otherwise.} \end{cases}$$

The specification should specify the target end state, the source of evidence, the quality threshold for the evidence, if required, the policy requirements and the deadline. A support task may need a verified resolution that doesn't have an account change that is prohibited, while a coding task may need an accepted change that has passed through the specified checks. Even if it looks useful, where the workflow result is not achieved, it is not considered successful.

NIST's AI Risk Management Framework considers issues of validity, reliability, safety, security and other properties of trustworthiness as contextual issues. This protocol is based on that and asks teams to identify the constraints that need to be met for success, and to report on other risk measures separately. Does not assume one universal successful test for products.

An automated judge could be used to evaluate an open-ended output but the decision rule must be validated with human or objective evidence. Biases, such as the position of the answer and verbosity, have been found in research on LLM judges.

4. Cost Boundary & CPST

4.1 Direct task cost

Suppose that execution cost of initiated task i is directly attributable to task i is denoted by C_i^D. Subject to documented prices and attribution rules, it comprises:

$$C_i^D = C_i^M + C_i^T + C_i^I + C_i^V + C_i^H + C_i^R,$$

where M represents model usage, T represents paid tools and external services, I represents measurable runtime of infrastructure that can be attributed to the workflow, V represents verification, H represents measurable human work in the workflow, and R represents retries, fallbacks and recovery actions. These categories refer to the source of the costs and

should not be double counted. For instance, a retry's model charge is only logged in model usage or in retry cost, not in both.

Human work is review and corrections and/or approval or escalation; if this work is part of the defined path to completion. It is priced based on the time recorded and a laid out labor-cost assumption. Due to the significant impact on comparison, human minutes and the percentage of tasks that need to be intervened should also be reported.

All engineering, procurement, security, evaluation-platform and governance costs that are not variable with the number of users should be allocated in a separately identified fully loaded view, unless a justifiable allocation method is identified. Direct CPST should be differentiated from downstream business loss, customer harm and opportunity cost. They could be very important and if they are added to observed execution charges, then the base measure becomes more difficult to reproduce.

4.2 Estimator

The proposed operational measure is: for N tasks that were initialized, measured under a single workflow policy and time window, the number of tasks that are eligible.

$$\widehat{\mathrm{CPST}}_{\mathrm{D}} = \frac{\sum_{i=1}^{N} C_i^{D}}{\sum_{i=1}^{N} S_i}, \qquad \text{when} \sum_{i=1}^{N} S_i > 0. \tag{1}$$

All finished eligible tasks, even failed, abandoned and timed out tasks, add their execution cost. Tasks that pass the prespecified success rule count towards denominator. If none of the tasks is successful, then the observed ratio is undefined and the report should be "0 successes, total expenditure, sample size" and not "infinite".

This is a portfolio accounting ratio equation (1). It's not the average cost of completed jobs, or some assurance that a specific unsuccessful request can be retried until it is successful. The population analogue of a stable population with a finite expected cost and a positive rate of success is:

$$\mathrm{CPST}_{\mathrm{D}} = \frac{\mathbb{E}[C^{D}]}{\Pr(S = 1)}. \tag{2}$$

This relation derives from a sum of costs and successes over all tasks that were initiated. Does not need independent attempts inside of a task. It does depend on a well-defined policy of the population and the process for the interpretation to be valid.

A fully loaded measure can have a disclosed allocation of costs of shared and fixed costs over the same period, A:

$$\widehat{\mathrm{CPST}}_{\mathrm{FL}} = \frac{\sum_{i=1}^{N} C_i^{D} + A}{\sum_{i=1}^{N} S_i}. \tag{3}$$

The direct and fully loaded measures serve different decisions and should be labeled separately.

5. Reliability, Retries and Service Constraints

If the cost of each attempt is the same (independent of all other attempts), if the probability of success is p>0, and if the number of attempts is unlimited and the success is recognized immediately, the expected cost of this process until success is achieved is c/p. This is not the cost-of-pass setting, but rather is a general model of production agent workflows.

If a workflow is allowed at most K attempts, and the attempt j has a conditional success probability p_j (and a cost of c_j if executed), then the expected cost and the eventual success probability of the workflow is:

$$\mathbb{E}[C] = \sum_{j=1}^{K} c_j \prod_{h=1}^{j-1} (1 - p_h), \tag{4}$$

$$\Pr(S = 1) = 1 - \prod_{j=1}^{K} (1 - p_j). \tag{5}$$

If such conditional probabilities and costs are sufficient to model the retry policy, they can be used in equations (4)–(5). Actual workflows can have different prompts or models depending on the workflow, can fail in dependent ways, may have variable tool charges or can escalate to a person. If that is the case, then the measured costs and outcomes for Equation (1) should be used in those cases, and the simplified retry equations are not to be used in place of the measured data.

Where decisions are required with a response time or other service requirements, success is also not enough. If a task has to be completed within deadline L, then let $S_i^{L} = 1$if the task meets the outcome criterion and L. Report CPST in conjunction with the underlying success rate, the latency distribution and the intervention rate. This makes the impact of the deadline apparent, without having to give an arbitrary monetary value to each minute of latency.

A candidate system is dominated for a given workload if there is another system with a lower direct CPST, with a higher success rate when the same workload is used (assuming the success rate is a function of the workload), and with a worse stated service constraint that includes at least one strict improvement. When there are no obvious system-based dominants, the product team needs to specify the acceptable cost, quality, latency and risk levels. This is a means of deciding which of the models is better, not proof that one family of models is "truer" than the other.

6. Measurement and Reporting Protocol

If two CPST results are referred to as comparisons, then the following items should be published prior to rankings:

Item	Required specification
Task population	Eligibility rules, source, period, exclusions and task categories
Outcome	End-state test, quality threshold, policy gates, time, and adjudication method
Workflow	Model and Tool versions, prompts, Retry limits, Fallback rules, Human escalation policy
Costs	Price basis, currency, measurement period, included categories and shared-cost allocation
Attribution	Task identifier, step identifiers, retry relationships, and rules to prevent double counting
Reporting	Initiated tasks, successes, total cost, CPST, first pass success, human time, latency, and missing data

The Minimum task record includes the following: Task identifier, task category, workflow version, start and end times, final outcome, why it failed or escalated, model and tool usage, measured human minutes, and attributed cost. Step level traces should indicate whether it is an original action or a retry/fallback. OpenTelemetry already has a set of conventions for generative AI use cases, like model usage, tool calls, evaluation scores, and workflow identifiers. Task-level cost accounting could be done with these signals in OpenTelemetry without claiming to create tracing.

Any tasks that have not had costs and/or have not had outcomes should be reported and given a treatment. Analysts should report the amounts of charges observed on the invoice or estimated based on published charges, or allocated from shared infrastructure. These are evidence types that are in fact materially different.

7. Empirical Evaluation Design

The protocol can test a claim: task-level accounting can alter the ranking of candidate agent systems over and above a token/inference price ranking. Previous cost-of-pass research, however, has already revealed that the fact that the tokens are of low value does not necessarily mean that they have low cost per correct solution; therefore, an empirical study of this protocol should reveal further benefits in terms of the workflow's attribution and the expected outcomes.

If a study is to be done, it would be done using the same prespecified task population for each candidate workflow. It would fix the outcomes and record the number of calls to models, tools, retries, amount of human work performed, latency of the outcomes, and final state of each system and repeat stochastic executions as possible. Results are to be reported separately for the following categories of tasks, and combined for a declared workload mix.

The main analysis would involve ranking based on (i) model price or inference cost, (ii) direct CPST, (iii) fully loaded CPST in which allocations are credible. Pairing task level resampling can estimate uncertainty in differences between systems without obscuring the fact that the same tasks were attempted by systems. If the same type of task is performed over and over again, these should be grouped together when resampling or otherwise modeled as dependent observations. A study should also try to be sensitive to labor rates, shared cost allocation, success adjudication and workload mix.

If there is no stable decision-relevant distinction between the task-level costs and outcomes of the proposition and the costs of pass in the evaluated workflows, then there is not much support for the proposition. A reversal in ranking by itself would be ineffective if it occurs because of inconsistently defined tasks, labels for success that are not agreed on, or arbitrary cost allocations.

8. Limitations and Conclusion

CPST depends on a specific task population, workflow policy, success rule and cost bound. It is not able to create any business value without assistance. The customer experience, failure harm, and value of tasks completed can vary significantly between two systems with the same CPST. But a poor success rate can be camouflaged by a low CPST when only the ratio is reported. For these reasons, CPST needs to be included with its denominator, success rate, latency, human intervention and relevant risk measures.

In this paper, only a suggested measurement protocol is proposed. It has not captured any task traces collected, comparative experimental, validated cost-attribution implementation or measured improvement in product decisions. The next research step is to test the protocol across a number of agent workflows and see if the extra accounting rules are reproducible and have a material impact on decisions.

Cost per successful task is best conceptualized as an application at the task level of established outcome based economics. The potential value of the research is the explicitness of the unit, outcome, boundary and attribution rules, so that other people can measure and challenge them.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Erol, M. H., El, B., Suzgun, M., Yüksekgönül, M., and Zou, J. (2025; revised 2026). "*Cost-of-Pass: An Economic Framework for Evaluating Language Models*." arXiv:2504.13359.

[2] FinOps Foundation. "*FinOps for AI: Tools & Services Considerations*." Accessed September 2026.

[3] FinOps Foundation. "*Unit Economics*." FinOps Framework. Accessed September 2026.

[4] Jimenez, C. E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O., and Narasimhan, K. (2023). "*SWE-bench: Can Language Models Resolve Real-World GitHub Issues?*." arXiv:2310.06770.

[5] National Institute of Standards and Technology. (2023). "*Artificial Intelligence Risk Management Framework 1.0: AI Risks and Trustworthiness*."

[6] OpenTelemetry. "*Generative AI Semantic Conventions*." Accessed September 2026.

[7] Wang, N., Hu, X., Liu, P., Zhu, H., Hou, Y., Huang, H., Zhang, S., Yang, J., Liu, J., Zhang, G., Zhang, C., Wang, J., Jiang, Y. E., and Zhou, W. (2025). "*Efficient Agents: Building Effective Agents While Reducing Cost*." arXiv:2508.02694.

[8] Yao, S., Shinn, N., Razavi, P., and Narasimhan, K. (2024). "*τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains*." arXiv:2406.12045.

[9] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. (2023). "*Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*." arXiv:2306.05685.