

PRACTITIONER GUIDANCE

AI Capitalization Guidance

Tracking and attributing AI development costs

A practical framework across enterprise tracking-maturity levels

BY

Joshua Roberto
TBM & FinOps Lead
Yarken

Be ready for the age of AI.

One IT spend system. Every dollar. Full visibility. Total control.

Executive Summary

AI is becoming one of the fastest-growing categories of technology investment, with spending accelerating rapidly as organizations expand adoption across software, infrastructure, models, and services. Gartner's 2026 AI Spending Forecast forecasts worldwide AI spending to reach \$2.59 trillion in 2026, a 47% year-over-year increase, with AI software spending rising from about \$283 billion in 2025 to \$453 billion in 2026.

Most organizations have mature processes for capitalizing software development costs, but very few have established governance for AI tools, AI agents, LLM usage, and AI-assisted software development.

Without project-level visibility into AI costs, Finance teams face increasing difficulty:

- Identifying eligible capitalization opportunities
- Supporting audit evidence
- Measuring AI ROI
- Forecasting AI investments
- Governing enterprise AI adoption

As AI investment accelerates, organizations that cannot identify capitalizable AI costs may be leaving millions of dollars on the table each year.

AI Economics provides the visibility needed to distinguish operational AI expenses from long-term strategic investments.

AI development tools introduce new considerations for software capitalization, but there are currently no accounting standards specifically governing AI development costs. Instead, organizations must apply existing accounting guidance based on the nature and intended use of the underlying software or activity. Internal-use software generally falls under ASC 350-40, software developed for sale or licensing under ASC 985-20, and certain R&D activities under ASC 730. For internal-use software, the existing ASC 350-40 framework applies until ASU 2025-06 is adopted, after which the updated recognition criteria apply. AI does not change the underlying accounting principles; it introduces new

cost types and attribution challenges that must be evaluated within those existing frameworks. Organizations therefore need appropriate processes and evidence to support the nature, timing, attribution, and accounting treatment of AI-related costs.

The most common mistake is assuming that how an AI tool is priced determines whether it can be capitalized, that a metered, usage-based charge is automatically capitalizable, or that a flat subscription fee is automatically overhead. Neither is true. Pricing structure is irrelevant to the capitalization test; direct attribution and supportable evidence are what matter. This guidance sets out the operative test, a clean capitalizable/expensed reference, and three fully worked scenarios corresponding to the three tracking-maturity levels most enterprises will find themselves in.

As organizations invest millions in AI initiatives, many continue to expense costs that may qualify for capitalization under applicable accounting policies. AI-enabled software development, internally developed AI models, implementation efforts, and certain development activities may be eligible when established accounting criteria are met. Without clear visibility into these investments, eligible costs can be overlooked and valuable assets may never appear on the balance sheet, creating missed capitalization opportunities that can grow into millions of dollars annually as AI becomes a larger portion of technology spend.

Why This Can't Wait

Consistency in treatment is priority #1.

(Consistent accounting is right accounting)

The right time to establish a capitalization methodology is before AI spend accelerates further, not after. Changing the accounting treatment once spend has already scaled, or revisiting prior periods to correct inconsistencies, creates a far more difficult conversation than establishing a clear, consistent approach from the outset.

Get the treatment right from the beginning, and there is nothing to correct later. The longer you wait, the greater the potential adjustment or missed capitalization becomes as AI investment scales.

An Illustrative Five-Year View

	TODAY	YEAR 5 (ILLUSTRATIVE)
Total annual AI spend	\$18.4M	\$150M – \$200M
Capitalization-eligible share (~11%)	\$2.1M	\$17M – \$22M
If the tracking gap isn't closed	Fully expensed today	Fully expensed at 8–10x today's dollar impact

Figures are illustrative: the spend trajectory assumes today's growth rate moderates over time, and the eligible share is held at today's approximate ratio for comparison purposes. The point isn't the precision of the projection, it's that the gap between what's eligible and what's actually captured doesn't shrink on its own. It scales with the spend.

AI is already the fastest-growing line in most technology budgets. Building this tracking capability now costs little relative to retrofitting it once the numbers are five to ten times larger, and considerably less than explaining an inconsistent methodology to an auditor after the fact.

The risk here isn't that today's AI spend is being treated incorrectly in isolation, it's that without a tracking mechanism, the treatment can't be verified either way, and that gap gets more expensive to close the longer it's left open.

Regulatory Foundation

Two related bodies of guidance govern the scenarios in this document:

STANDARD	WHAT IT MEANS FOR AI COSTS
ASC 350-40 Internal-use software	Costs are capitalizable only during the application development stage, after the preliminary/research stage has concluded and management has authorized funding, and only for external direct costs of materials and services, plus payroll and payroll-related costs for employees directly associated with the project.
ASU 2025-06 Probable-to-complete threshold	For entities that have adopted it, the capitalization window opens once it becomes probable the project will be completed and the software will perform its intended function which may land later than the original authorization date if significant technical uncertainty remains open.

Post-deployment costs, bug fixes, monitoring, and routine maintenance fall back to expense treatment regardless of which framework applies. AI tooling costs are evaluated inside these same boundaries; they do not get a separate standard.

The Core Capitalization Test

Every AI-related cost line such as labor, subscription fees, token/API usage, and compute should be run through the same three-part test. All three conditions must be met; failing any means the cost is expensed.

CONDITION	QUESTION TO ANSWER
1 Scope	Is this resource dedicated to one identified, capitalization-eligible project, with access restricted to that project's team? Broadly available resources fail here, regardless of how they're priced.
2 Activity	Within that dedicated resource, was the work build-and-test activity (code generation, feature development, testing) or research, maintenance, production support, or training?
3 Evidence	Can the cost of the qualifying activity be isolated with task- or project-level records, rather than inferred from aggregate usage or a headcount-based estimate?

Pricing structure does not determine capitalizability

A token-based charge is not automatically capitalizable simply because it is measurable, and a fixed subscription is not automatically an overhead cost simply because it isn't usage-metered. A project-dedicated, access-restricted subscription can be fully capitalized; a broad enterprise license used across finance, marketing, and multiple engineering teams cannot, even though the reverse pricing intuition is common in practice.

The mixed-activity session

The hardest real-world case is the mixed-activity session where a single AI agent session that does qualifying work (generating code for an authorized project) and non-qualifying work (investigating a production incident, general research) back-to-back. Absent task-level evidence to split the session, the entire session may need to be expensed; partial project relevance does not entitle partial capitalization without a defensible allocation basis.

Capitalizable vs. Expensed AI Costs

CAPITALIZABLE

- Developer/engineer time spent coding, refactoring, or testing with AI assistance during the active development stage
- Fixed license fees or metered usage credits that are tracked and restricted to an identified project build
- Model fine-tuning, validation, and training-run costs when performed as the direct equivalent of coding/testing a specific internal-use asset

EXPENSED AS INCURRED

- General productivity use or broad exploratory use that isn't tracked to a specific project
- Preliminary-stage work: feasibility studies, vendor bake-offs, architecture exploration before a project is authorized
- AI-assisted bug fixing, routine maintenance, or monitoring after deployment
- Staff training on how to use AI tools

Three Scenarios by Enterprise Tracking Maturity

Not every organization has the same level of AI usage instrumentation. The methodology for attributing cost should match what your tooling can actually support. From fully metered, project-tagged usage down to teams with no timesheets and no token-level metering at all. The three scenarios below correspond to three maturity levels enterprises most commonly encounter, each with a fully worked calculation and journal entry provided as an example.

Each scenario below represents a single project, a single tool, or a single pooled license; one component of a much larger AI spend portfolio. At that scale, the dollar amounts involved are not material to a large enterprise on their own. Applied consistently across the full portfolio, rather than to a single illustrative project, the same math scales quickly. The individual scenario is the unit of measurement, not the total opportunity; the materiality shows up when the methodology is applied everywhere AI spend already exists, and compounds further as that spend expands in the years ahead.

MATURITY	WHAT YOU HAVE	METHODOLOGY
Scenario 1	Metered, project-tagged usage	Direct tracing
Scenario 2	Pooled license + individual timesheets	Timesheet-based proxy
Scenario 3	Pooled license + dedicated agile team, no timesheets	Agile-output ratio proxy

SCENARIO 1

Metered, Project-Tagged Usage

Highest maturity

Applicable when AI tool usage can be traced to a specific repository or project via API tagging, and developers log hours across multiple projects.

PROJECT ALPHA — CORP PRODUCT

- Authorized January 1; launched June 30.
- Capitalization window: March 1 – May 31 (3 months). January–February was spent resolving architectural uncertainty (preliminary stage — expensed); June was spent on post-launch bug fixes (maintenance stage — expensed).
- Tooling: Cursor Teams at \$40/seat/month, plus metered overages once included credits are depleted, tracked to specific repositories via an API gateway.

Tool Cost Allocation

COMPONENT	BASIS	AMOUNT
Flat license fee	500 developers × \$40 × 3 months, all working exclusively on Project Alpha	\$60,000
Metered overages	\$180,000 total company-wide × 80% traced to Repo-Alpha	\$144,000
Total capitalizable tool cost		\$204,000

Labor Cost Allocation — Developer Jane Doe (\$26,000/month fully burdened)

METRIC	VALUE
Total hours logged, Mar–May	480 hours
Hours on Project Alpha (active coding/testing)	360 hours
Capitalization %	$360 / 480 = 75\%$
Total 3-month labor cost	$\$26,000 \times 3 = \$78,000$
Capitalizable labor cost	$\\$78,000 \times 75\% = \\$58,500$

Journal Entry — Project Alpha (Jane Doe's portion)

DEBIT Internal-Use Software (Asset) \$262,500 (\$204,000 tool + \$58,500 labor)

CREDIT AI Tooling Vendor Expense (P&L) \$204,000

CREDIT Software Engineer Payroll Expense (P&L) \$58,500

Project Alpha's full build team includes 500 developers on this same tagging and timesheet basis. Extrapolated across the team, this quarter's total capitalizable spend on the project is approximately \$29.5M (\$204,000 tool + 500 × \$58,500 labor ≈ \$29,454,000).

SCENARIO 2**Pooled License, Individual Timesheets Available**

Mid maturity

Where direct project-level AI consumption data is unavailable, documented developer hours may provide a reasonable and supportable allocation driver for pooled AI costs when those hours are demonstrably correlated with use of the shared AI resource. The methodology should be consistently applied, reconciled to the underlying pooled cost, limited to qualifying development activities, and documented within the organization's capitalization policy.

PROJECT HORIZON — INTERNAL HR TALENT EVALUATION SYSTEM

- Pooled enterprise AI license: \$5,000/month flat for a block of seats; token-level usage is not tracked or separable by repository.
- Budget authorized April 1. Under ASU 2025-06, the “probable-to-complete” threshold was not met until May 1, once a significant technical uncertainty (data-privacy compliance under load) was resolved through initial testing. Software launched June 30.
- Capitalization window: May 1 – June 30 (2 months). April is 0% capitalizable despite active work, because the completion threshold had not yet been met.

Developer Alex — Time Log and Capitalization Status

MONTH	TOTAL HOURS	HOURS ON HORIZON	STATUS
April	160	120	0% — blocked by development uncertainty
May	160	140	Eligible — threshold met (87.5%)
June	160	80	Eligible — prior to launch (50.0%)

Allocation — Alex is 1 of 25 seats on the \$5,000/month pooled license (\$200/seat)

MONTH	TOOL COST ALLOCATED	LABOR ALLOCATED (\$15,000/MO)
May (87.5%)	$\$200 \times 87.5\% = \175	$\$15,000 \times 87.5\% = \$13,125$
June (50.0%)	$\$200 \times 50.0\% = \100	$\$15,000 \times 50.0\% = \$7,500$
Total	\$275	\$20,625

Journal Entry — Project Horizon (Alex's portion)

DEBIT	Internal-Use Software (Asset) \$20,900 (\$275 tool + \$20,625 labor)
CREDIT	Shared AI Software Subscription Expense (P&L) \$275
CREDIT	Engineering Payroll Expense (P&L) \$20,625

This approach provides a clear and supportable basis for audit review. April costs were excluded because the capitalization window is determined by the completion threshold rather than the presence of active work. Similarly, tool costs are allocated based on Alex's documented output hours, providing a more representative allocation than a standard per-seat approach.

SCENARIO 3**No Metering, No Timesheets, Dedicated Agile Team**

Most common state for mature agile organizations

Applicable when a team is 100% dedicated to one product and does not track individual hours — the most common state for mature agile organizations, and structurally identical to a Rally/story-based allocation approach used for traditional software capitalization.

PROJECT APEX — INTERNAL LOGISTICS OPTIMIZATION PLATFORM

- Dedicated cross-functional Scrum team of 6 engineers, 100% allocated to Project Apex; no weekly timesheets required.
- Tooling: GitHub Copilot Enterprise at a flat \$40/seat/month; usage metering is not enabled at the project or repository level.
- Active sprint cycles: March 1 – April 30 (2 months, four two-week sprints).

Agile Output Allocation — Story Points as the Capitalization Proxy

CATEGORY	STORY POINTS	SHARE
Capitalizable — new architecture, API connectors, UI features	180	75%
Expensed — bug fixes, tech debt, pipeline maintenance	60	25%
Total	240	100%

Cost Allocation at the 75% Ratio

COMPONENT	TOTAL INCURRED (2 MO)	CAPITALIZABLE (75%)	EXPENSED (25%)
AI tool cost (6 seats × \$40 × 2 mo)	\$480	\$360	\$120
Team labor (6 engineers, fully burdened)	\$180,000	\$135,000	\$45,000
Total	\$180,480	\$135,360	\$45,120

Journal Entry — Project Apex

DEBIT	Internal-Use Software (Asset)	\$135,360 (\$360 tool + \$135,000 labor)
CREDIT	Software Vendor Expenses (P&L)	\$360
CREDIT	Engineering Payroll Expenses (P&L)	\$135,000

This approach provides a strong, supportable basis for audit review by avoiding broad assumptions, such as capitalizing 100% of a dedicated team's costs simply because the team works on a single product. Instead, it uses objective, exportable sprint metadata such as story points tied to development epics, versus maintenance activities to appropriately account for non-development work. The methodology should be clearly documented within the organization's capitalization policy and applied consistently across reporting periods.

Governance Policy Checklist

- Map every AI cost line: labor, subscription, metered usage, and compute to a named, authorized project before it is capitalized.
- Confirm the capitalization window against the applicable threshold: active development stage under ASC 350-40, or the probable-to-complete date under ASU 2025-06.
- For metered usage, configure tagging at the API/gateway level so every request carries a project billing code; do not rely on manual reclassification after the fact.
- For timesheet-based allocation, ensure logged hours are tied to task-level categories (project build vs. legacy/maintenance), not a single generic project bucket.
- For agile teams without timesheets, use an objective, consistently applied engineering metric (story points, accepted-work counts) split between capitalizable and non-capitalizable ticket types, never a flat 100% claim for a dedicated team.
- Never allocate a broad, multi-project AI license by headcount as a substitute for direct attribution, headcount-based spreading does not convert an indirect cost into a direct one.
- Preserve the underlying artifacts (API tag logs, timesheet exports, sprint/velocity reports) as the supporting evidence file for each capitalized amount, not just the final journal entry.

Recommended Implementation Path

Identify which of the three scenarios matches each team's current tracking maturity, then apply the corresponding methodology consistently:

- Metered + project-tagged usage → use direct tracing (Scenario 1).
- Pooled license + individual timesheets → use the timesheet-based proxy (Scenario 2).
- Pooled license + dedicated agile team, no timesheets → use the agile-output ratio proxy (Scenario 3).

Whichever methodology applies, route the tagging taxonomy and allocation logic through a governance review with Accounting Policy and IT Finance before go-live, and pilot it in parallel against the prior costing method for at least one full reporting cycle to confirm the results are consistent and defensible before relying on it in the general ledger.

Accounting Policy Disclaimer

This document is intended to provide a practical framework for identifying AI-related costs that may qualify for capitalization under applicable accounting guidance. Final capitalization decisions should be made by the organization's Accounting or Finance team based on its accounting policies, materiality thresholds, and consultation with external auditors where appropriate.

References

FASB ASC 350-40, Internal-Use Software – Codification (asc.fasb.org)

FASB ASU 2025-06, Targeted Improvements to the Accounting for Internal-Use Software

Deloitte, “Accounting for AI Costs Associated with Internal-Use Software Development,” Technology Spotlight, 2026 (dart.deloitte.com)