



In This Issue

- [Executive Summary](#)
- [Overview](#)
- [Background](#)
- [Relevant Accounting Principles](#)
- [A Framework for Evaluating AI-Related Costs](#)
- [Contacts](#)

Accounting for AI Costs Associated With Internal-Use Software Development

Executive Summary

- In developing internal-use software, companies are increasingly using AI-enabled tools and services alongside internal employees and third-party software developers. As AI agents perform tasks historically performed by people — such as planning development work, generating or reviewing code, executing tests, debugging, preparing documentation, and interacting with development tools — questions have arisen about how to account for the associated technology costs. To account for these costs, companies must use judgment in applying existing guidance on capitalization and expense.
- The internal-use software accounting framework does not change merely because the development resource is an AI-enabled service rather than a person. However, the nature, pricing, and evidence of activities incurring the costs may have changed. Historical processes and controls used to track and attribute personnel costs may not be sufficient for AI-related costs, particularly when AI services are shared among users, projects, environments, or business functions. Entities may therefore need to revise their cost-accounting processes, how they set up projects to link to AI development activities usage reporting, and controls.
- The accounting does not depend on the form of the pricing arrangement. A token-based charge is not capitalizable merely because it is measurable, and a fixed or subscription-based charge is not necessarily noncapitalizable merely because it is not billed by usage. The analysis should focus on the nature of the service, how and when the service is consumed, the activity supported, and whether the associated cost can be directly attributed to a qualifying software project.¹

¹ A software project that has met the recognition threshold for capitalization.

Overview

The software development model is shifting from a predominantly human-driven approach to a blended model that combines human effort with AI-enabled services. Contemporary AI development tools may assist with or autonomously perform portions of the software development workflow, including technical planning, code generation, testing, debugging, code review, documentation, and other development activities. This *Technology Spotlight* discusses key considerations in the assessment of whether companies should capitalize AI-related costs in developing internal-use software or expense such costs as incurred.

Background

Historically, human effort has been the primary factor affecting the cost of developing internal-use software. To support capitalization, many entities established processes to track the activities' time and costs associated with internal employees and external contractors and to attribute those costs to defined software projects. Those processes generally relied on a direct link between an identifiable person, the time spent, the project on which he or she worked, and the nature of the activity performed.

AI-enabled development tools disrupt several of the assumptions underlying that model. The same tool is often available to a developer for all of his or her work — new development, maintenance, production support, research, and general productivity — rather than being dedicated to a single project. As a result, the traditional link between a cost, a specific project, and an activity is frequently absent, and an entity cannot assume that a cost is related to an activity that supports capitalization simply because it was incurred by a member of a development team.

This challenge is compounded by how AI tools operate. A single AI coding agent may be used to perform multiple activities within a single session — for example, inspecting a repository, preparing a development plan, generating code, running tests, diagnosing errors, revising the code, and preparing a pull request — while in the same session also researching alternatives or investigating a production issue. Activities that support capitalization may therefore be intermingled with those that do not in a single session in such a way that the tool being used does not, by itself, indicate whether the underlying activity meets the capitalization criteria.

The economics have also changed. Rather than the labor-based costs entities are accustomed to tracking, AI tools may be obtained through per-user or organization-wide subscriptions, token or application programming interface (API) charges based on interactions with an AI model; task- or compute-based pricing; committed-consumption arrangements; dedicated computing capacity; fixed-price contracts for specified development services; or arrangements containing both fixed and variable charges. Many of these arrangements meter usage through consumption measures. A token, for example, is a unit of information processed by an AI model and may represent a portion of the text, code, image, or other information provided to or generated by the model; other services meter usage through API calls, agent tasks, compute time, or credits. These measures can provide useful evidence about the amount and timing of AI services consumed, but the existence of a usage measure does not govern whether a cost is related to an activity that supports capitalization.

These developments do not change the accounting framework, but they do change what an entity must be able to identify and support related to capitalized costs. The questions below — what activity the AI service performed, what the entity is paying for, and whether the resulting qualifying development activity cost can be identified and supported — frame the analysis in the remainder of this publication.

Relevant Accounting Principles

ASC 350-40-30-1² states, in part:

Costs of computer software developed or obtained for internal use that shall be capitalized include only the following:

- a. External direct costs of materials and services consumed in developing or obtaining internal-use computer software. Examples of those costs include but are not limited to the following:
 1. Fees paid to third parties for services provided to develop the software during the application development stage
 2. Costs incurred to obtain computer software from third parties
 3. Travel expenses incurred by employees in their duties directly associated with developing software.
- b. Payroll and payroll-related costs (for example, costs of employee benefits) for employees who are directly associated with and who devote time to the internal-use computer software project, to the extent of the time spent directly on the project. Examples of employee activities include but are not limited to coding and testing during the application development stage.

Under ASC 350-40, external direct costs of materials and services consumed in developing or obtaining internal-use software must be capitalized when the applicable capitalization requirements have been met. Such costs may include fees paid to third parties to develop software, costs of software obtained from third parties, and other third-party materials or services consumed in developing the software. Accordingly, for the capitalization requirements to be met, the cost should be more than generally related to the software development life cycle. Specifically, the cost should:

- Represent an external cost of materials or services consumed in developing or obtaining internal-use software.
- Be directly attributable to a specific qualifying software project or projects.
- Be related to a qualifying development activity.³
- Be incurred during the period in which costs for the project are eligible for capitalization.

Payroll and payroll-related costs for employees who are directly associated with an internal-use software project may also be capitalized to the extent that the employees spend time directly on qualifying project activities. General and administrative costs and overhead are expensed as incurred. Training, maintenance, and data-conversion costs are also generally expensed.

Fees paid to third parties for use of AI do not automatically constitute a new category of capitalizable cost. The same underlying principles that apply to other internal and external development resources should be used to evaluate AI-related costs. The accounting should reflect the substance of the services obtained rather than the technology used to provide them.

A cost also does not become a direct project cost merely because it is assigned to a software project through an internal cost allocation process. Allocating the cost of a broadly available AI tool to software projects on the basis of developer headcount, labor hours, broad project budgets, or similar measures does not, in and of itself, establish that the cost is related to materials or services directly consumed by those projects.

² For titles of *FASB Accounting Standards Codification* (ASC) references, see Deloitte's "[Titles of Topics and Subtopics in the FASB Accounting Standards Codification](#)."

³ Capitalizable software development activity.

However, this principle does not preclude the use of operating data to measure an otherwise direct cost. For example, when a vendor invoice includes identifiable API services consumed by software projects and activities performed, these reliable usage records may be used to determine the cost of the services consumed by each project. In that situation, the entity is measuring direct service consumption, not converting an indirect cost into a direct cost through general allocation.



Connecting the Dots

In September 2025, the FASB issued [ASU 2025-06](#)⁴ to amend its guidance on internal-use software. The amendments remove references to discrete software development project stages and establish a recognition threshold under which capitalization begins when:

- “Management, with the relevant authority,” has authorized and committed to the funding of the software project.
- “It is probable that the project will be completed and the software will be used to perform the function intended.”

In determining whether completion of the project is probable, an entity considers whether “significant uncertainty [is] associated with the development activities of the software.”

The amendments do not change the types of internal-use software costs that are eligible for capitalization, the need to establish the software’s intended function and system requirements before a cost is eligible for capitalization, or when capitalization ceases. A cost continues to only be eligible for capitalization after performance requirements are established, and capitalization ceases when the software is substantially complete and ready for its intended use.

The amendments are effective for annual reporting periods beginning after December 15, 2027, including interim periods within those annual periods. Early adoption is permitted. Before adopting the amendments, an entity should continue to apply the existing project-stage recognition model.

Accordingly, an entity evaluating AI-related costs should determine whether it is applying (1) the existing ASC 350-40 project-stage model or (2) the amended recognition model under ASU 2025-06.

This distinction affects when capitalization begins. It does not change the analysis of whether an AI-related cost represents an external direct cost of materials or services consumed in developing internal-use software.

For more information about ASU 2025-06, see Deloitte’s September 18, 2025, [Heads Up](#).

A Framework for Evaluating AI-Related Costs

The accounting for an AI-related cost depends on the activity performed, the nature of the service or resource the entity is paying for, and whether the cost of qualifying development services can be directly attributed to an eligible software project. The questions below may provide entities with a framework for applying those principles.

⁴ FASB Accounting Standards Update (ASU) No. 2025-06, *Intangibles — Goodwill and Other — Internal-Use Software (Subtopic 350-40): Targeted Improvements to the Accounting for Internal-Use Software*.

1. What Activity Does the AI Service Perform?

AI offerings may provide code generation, code review, repository search, agent orchestration, automated testing, data processing, computing capacity, documentation support, production troubleshooting, or general research functionality. A vendor's description of a product as an "AI coding assistant" or "coding agent" does not govern the accounting, because the same product may support both capitalizable and noncapitalizable activities under ASC 350-40.

Provided that the applicable recognition requirements are met, AI services used to generate or modify code for approved functionality, develop interfaces, configure software, perform development testing, or correct defects before the software is ready for its intended use may be capitalizable development activities.

Costs associated with AI services used for preliminary evaluation, general experimentation, employee training, routine maintenance, production support, or general research and productivity are not capitalizable under ASC 350-40 merely because they are used by members of a software development team.

A single AI agent may perform multiple activities during a single session. For example, an agent may research alternative technical approaches, generate and test code, prepare training materials, and investigate a production issue. As with costs for an employee that incurred time on all those activities, the entire session should not be treated as capitalizable merely because part of the session is related to software development.

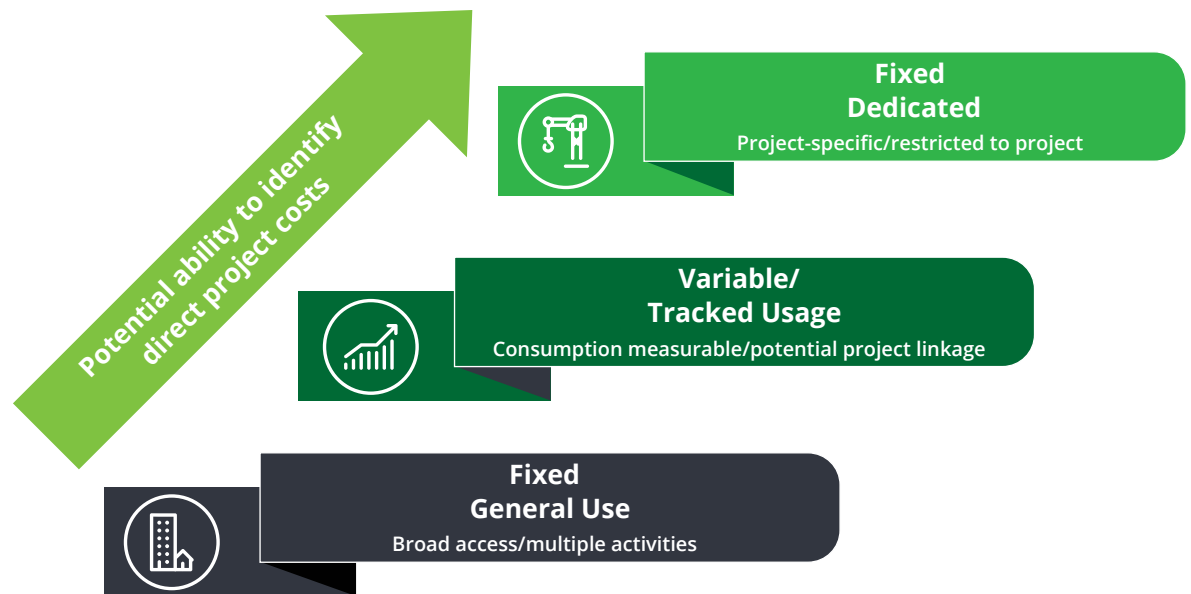
When reliable information provides a supportable basis for distinguishing services associated with capitalizable activities from those that are not, only the costs associated with the capitalizable activities should be considered for capitalization. If there is no supportable basis for distinguishing those services, capitalization of the combined cost may not be appropriate.

When evaluating AI service costs associated with software development, management should understand the types of activities that will be performed by the AI services purchased as well as how the software development team uses the AI offering.

2. What Is the Entity Paying For?

AI-enabled tools and services may be obtained through fixed subscriptions, usage-based charges, fixed-price development contracts, committed-consumption arrangements, reserved capacity, or combinations of those pricing models.

The accounting does not depend on the form of pricing. A fixed fee may represent a direct development service, while a measurable usage-based charge may be related to an activity not capitalizable under ASC 350-40. The analysis should distinguish between paying to make an AI tool available for use in an employee's work and paying for identifiable services or resources used to develop a particular software project.



Fixed Fee for an AI Tool Used for a Specific Project

An entity may purchase a fixed-term subscription to an AI coding tool specifically to develop an identified internal-use software project. If the subscription is obtained for that project, access is restricted to the project team, and the tool is used only in capitalizable development activities during the applicable capitalization period, the fee may represent an external direct cost of developing the software.

For example, an entity may purchase a six-month subscription to an AI coding agent solely for Project A. Access is limited to the Project A development team, and the tool is used to generate and review code, develop interfaces, and perform development testing for that project. The subscription fee may be capitalizable because the entity obtained the tool specifically for Project A, the cost can be directly identified with that project, and the cost is incurred during an eligible capitalization period. The fact that the fee is fixed and does not vary on the basis of tokens, prompts, or other usage measures does not, by itself, preclude capitalization.

Fixed Fee for an AI Tool Used Across Multiple Projects and Activities

A different conclusion may apply when a fixed subscription allows employees to use an AI coding tool for multiple projects and activities since such subscription may be more akin to a shared technology cost or overhead-like expense than a direct software-development cost. For example, developers may use the same subscription for new software development, maintenance, troubleshooting, research, documentation, production support, and general productivity (e.g., understanding existing code, explaining legacy application, meeting summaries and action items).

If the subscription fee is not directly associated with a particular software project, assigning the fee to projects on the basis of developer headcount, labor hours, broad project budgets, or similar internal measures does not make it a direct project cost. Such an assignment merely spreads the cost of providing access to the tool among projects.

The fact that a developer working on a software project in its capitalization period holds or uses the subscription does not, by itself, establish that the subscription fee is directly

attributable to that project. Similarly, an internal designation of a subscription as a project cost does not establish direct attribution if the tool remains available for broader use.

Usage-Based Charges

Charges based on tokens, API calls, agent tasks, or computing time may provide evidence of the amount and timing of AI services used. Measurable usage does not, however, establish that the associated activity is capitalizable under ASC 350-40.

For example, an entity's usage records may identify services used to generate and test code for Project A, maintain an existing application, and conduct general technical research. The charges associated with the capitalizable development activities for Project A may be capitalized if they are incurred during the applicable capitalization period. The charges associated with maintenance and research would not be capitalizable under ASC 350-40.

Unlike a general subscription fee, a usage-based charge may be directly identified with a specific project and activity. When it can, the associated cost may be capitalizable — not because usage was measured, but because the charge corresponds to an identifiable service used in a capitalizable development activity for an identified project.

Mixed Arrangements

Some contracts include a fixed fee and separate usage-based charges. Each component should be evaluated on the basis of what the entity receives.

For example, the fixed component may provide employees with access to an AI coding tool for multiple activities, while the variable component reflects project-level API usage. Alternatively, the fixed component may represent specified development services for an identified project.

The entity should not presume that all fixed charges are overhead or that all usage-based charges are direct costs. It should evaluate the substance of each component and the activities supported by the related service.

Prepaid and Minimum-Commitment Arrangements

An entity may pay up front or commit to a minimum payment for a specified quantity of tokens, credits, API calls, or other units of AI processing available during a contract term. The accounting depends on the substance of the arrangement, including whether the payment represents a prepayment for discrete units expected to be consumed or, instead, a fixed fee for access to AI services.

In the arrangements discussed below, the distinguishing principle is whether the per-unit charge reflects the direct cost of consumption. When the entity pays per unit consumed — or prepay for units it expects to consume substantially in full — measuring cost by units consumed reflects direct service consumption. When the payment is fixed and would be incurred regardless of how much is used in a project, assigning the payment on a per-unit basis represents allocation of a fixed cost and does not establish direct attribution. An entity must use judgment when expected consumption is uncertain.

Prepayment for Units Expected to Be Consumed

When an entity expects to consume the purchased units, the up-front payment generally represents a right to receive future AI services. The payment for usage credits is initially recorded as a prepaid asset and relieved as the AI services are received.

Credits consumed in capitalizable development activities for an identified internal-use software project during the applicable capitalization period may be capitalized if the associated cost is directly attributable to that project. Credits consumed for maintenance,

production support, research, training, general productivity, or other activities not capitalizable under ASC 350-40 should be expensed as the related services are received. Unconsumed credits generally remain recorded as a prepaid amount, subject to consideration of expiration or other applicable accounting requirements.

For example, assume that an entity prepays for 1 million AI credits and expects to consume all of them. During the month, 200,000 credits are consumed in capitalizable development activities for Project A, 100,000 credits are consumed in maintaining existing software, and 50,000 credits are used for general research. Subject to the applicable capitalization requirements, the cost of the 200,000 credits consumed by Project A may be capitalized. The costs associated with maintenance and research would be expensed, and the cost of the remaining 650,000 unconsumed credits would continue to be recognized as a prepaid amount. See additional discussion in [Example 7](#) below.

Minimum Commitment That Functions as a Fixed Fee

A different analysis may apply when the entity commits to a minimum payment but expects to consume substantially less than the included usage. In that circumstance, the payment may function economically as a fixed fee for access to AI services rather than as a prepayment for discrete units expected to be consumed.

For example, an entity may pay \$1 million for the right to consume up to 5 billion tokens during a one-year period but may expect to use only 600 million tokens and would never need 5 billion tokens in a period. If the entity does not expect to incur charges above the contractual minimum or expect to fully use the minimum, it will pay the same \$1 million regardless of whether it uses 400 million, 600 million, or 900 million tokens.

Token records may provide evidence of how the AI service is used; however, in this scenario, dividing the fixed payment — which is more akin to overhead — by the included tokens and assigning that amount to projects on the basis of relative token usage may not faithfully represent the cost of services directly attributable to those projects. If the included usage is available for multiple projects and activities, such a calculation may merely represent the allocation of a fixed overhead cost that the entity would incur regardless of the usage in any particular project. In determining the substance of the fixed commitment, entities will need to carefully evaluate minimum-commitment fixed payments, expected usage scenarios, and reasons for entering into a minimum commitment.

As discussed above, if the minimum-commitment arrangement is obtained specifically for an identified internal-use software project and may be used only in capitalizable development activities for that project, the minimum payment may represent a direct project cost. That conclusion would be based on the project-specific nature of the contracted AI service rather than on an allocation of the minimum payment to individual tokens consumed.

3. Can the Qualifying Development Activity Cost Be Identified and Supported?

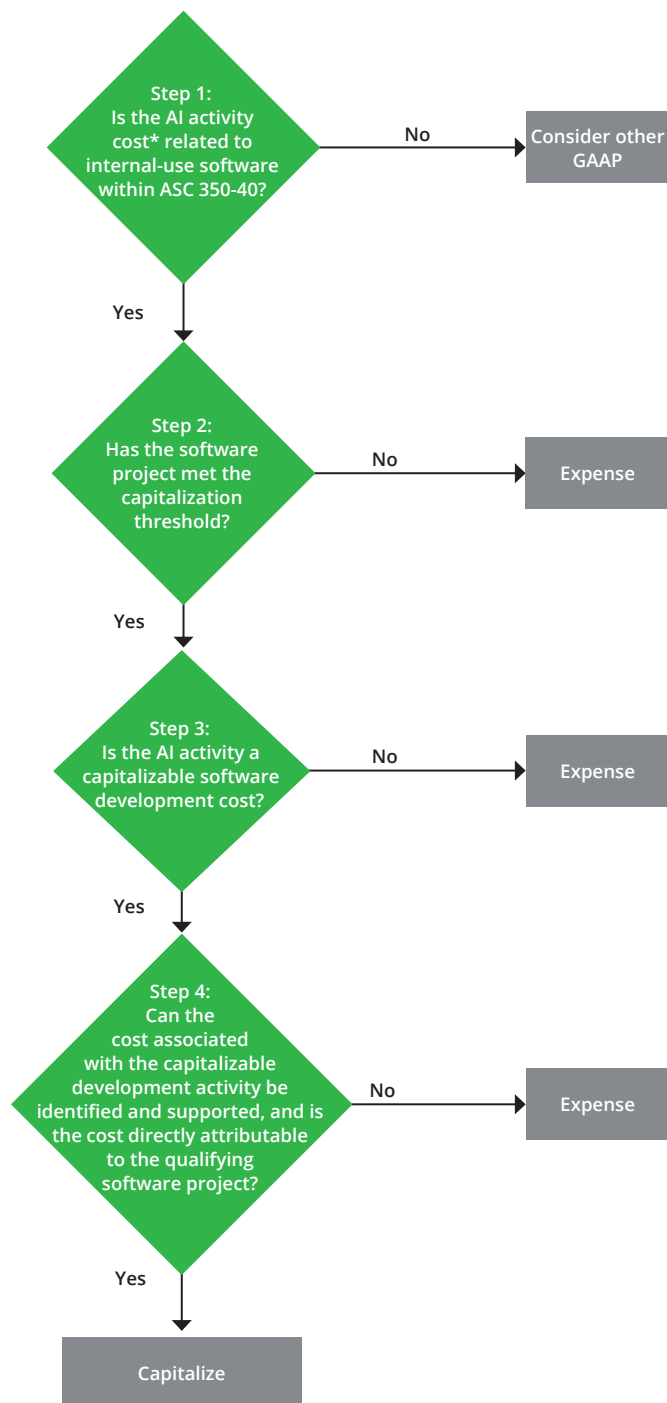
After identifying that the AI service will perform a qualifying development activity, the entity must determine whether the cost is directly attributable to the qualifying development activity, the software project associated with the activity, and whether the cost is supported by reliable evidence.

AI coding assistance may provide usage-based pricing models based on API calls, token consumption, or other usage metrics. As a result, entities may have access to significantly more detailed information regarding how AI services are consumed, including the user, time of use, AI model used, token consumption, and associated costs. Certain AI service platforms may also provide analytics and reporting capabilities that may enable organizations to monitor usage across teams, repositories, projects, or development environments.

The availability of usage data does not necessarily serve as sufficient evidence to identify costs eligible for capitalization. While token consumption and API activity may indicate the amount of AI resources used, they generally do not, on their own, demonstrate that the underlying activity was directly associated with a qualifying activity for a software project.

In evaluating whether AI coding-agent costs are eligible for capitalization, an entity should consider whether it has sufficiently detailed information to associate usage with specific software-development projects and activities. As with traditional labor-capitalization models, management may need processes and controls in place to identify the project benefiting from the AI service and to distinguish development activities from preliminary research, maintenance, training, and administrative activities. Sources of evidence may include AI-platform reporting or billing details, project-management systems, development tickets, and other project-accounting records. AI-platform details regarding who used the AI platform, when it was used, what was performed, and the associated cost may need to be combined with project-management records that provide additional details on activities occurring during the period of use.

Although AI service providers may furnish detailed reports regarding token consumption and API activity, entities may face practical challenges in using that information to identify project-specific costs. AI services are often used in multiple software-development projects and nonqualifying development activities, and a single user request may result in the use of multiple AI models, providers, or services before an output is returned. In addition, vendor usage information may not identify the specific project or activity associated with the underlying AI services. As a result, organizations may need to develop specific processes and controls to attribute token usage and associated costs to eligible software projects and activities.



* ASC 350-40 applies to costs incurred in developing or obtaining internal-use software. Accordingly, entities should evaluate the activities supported by the AI tool or service and exclude costs associated with non-software development activities before assessing capitalization.

Expensing AI Costs Associated With Internal-Use Software

Once an entity determines that AI-related software development costs do not meet the criteria for capitalization under ASC 350-40, those costs should be expensed as incurred. For costs that are expensed as incurred, an entity must determine the appropriate classification of those costs in the income statement on the basis of the function benefiting from the underlying activities. In a manner similar to attributing a cost to a specific project, the income statement classification of AI service cost will be in line with the nature of the activity it is related to, such as R&D, cost of goods sold, cost of services, or general and administrative activities. Organizations may need to develop processes, controls, and estimation methods to appropriately classify the costs incurred.

Illustrative Examples

The following examples illustrate how entities may apply the above considerations to AI cost arrangements used to develop internal-use software. These examples illustrate how differences in the activities performed, the nature of the services or resources obtained, the ability to directly attribute costs to a qualifying software project, and the scope of use may lead to different accounting outcomes.

Example 1

Entity A is developing a new internal-use financial reporting platform. As part of the development effort, A uses a third-party AI coding agent to generate and test application code.

The key terms and facts associated with the arrangement are as follows:

- The AI service is priced on the basis of token consumption.
- All token consumption can be tracked by project IDs, employees, and development environments.
- The AI agent is used exclusively by the engineering team assigned to the financial reporting platform during a period in which capitalization is appropriate.
- The AI agent is used to generate, refine, and test code that is incorporated into the financial reporting platform.
- The associated development effort has been approved, and significant development uncertainty has been resolved.
- Vendor-provided usage reports and system logs provide detailed evidence linking consumption directly to the project and person.

Because the token consumption is directly associated with a single internal-use software project, is related solely to qualifying development activities, and is supported by observable usage data, A concludes that the costs represent direct external costs of services consumed in developing internal-use software. Accordingly, the token-based charges are capitalized as part of the cost of the internal-use software under ASC 350-40.

Example 2

Entity B enters into an enterprise-wide arrangement with a third-party AI provider that offers employees access to a suite of AI capabilities, including coding assistants, document generation tools, and data analysis functionality.

The key terms and facts associated with the arrangement are as follows:

- The arrangement is priced on a fixed, per-seat monthly subscription basis.
- Access is provided broadly across the organization, including engineering, finance, marketing, and operations.
- Engineering personnel use the AI tools to assist with certain coding activities; however, the same tools are also used for general productivity purposes, such as drafting communications, researching existing code, and preparing analyses.
- The arrangement provides continuous, stand-ready access to AI capabilities regardless of actual usage.

Because the arrangement provides broad, enterprise-wide access to AI capabilities, supports multiple functions and activities at the organization, and is not attributable to a specific internal-use software project or qualifying development activity, B concludes that the costs represent general-purpose tools and shared overhead. Accordingly, the subscription fees are expensed as incurred.

This conclusion reflects the nature of the arrangement rather than the fact that it is priced on a fixed, per-seat basis. As illustrated in [Example 3](#), a fixed subscription obtained specifically for, and restricted to, an identified project may support a different conclusion.

Example 3

Entity E purchases a fixed-term subscription to an AI coding agent specifically to develop Project A, an identified internal-use software project.

The key terms and facts associated with the arrangement are as follows:

- The subscription is a fixed, six-month fee that does not vary on the basis of tokens, prompts, or other usage measures.
- The subscription was procured solely for Project A, and access is restricted to the Project A development team.
- The tool is used to generate and review code, develop interfaces, and perform development testing for Project A during a period in which capitalization is appropriate.
- The subscription is not made available for other projects, maintenance, production support, or general productivity.

Because E obtained the tool specifically for Project A, restricted access to the project team, and used the tool only in qualifying development activities during the capitalization period, E concludes that the subscription fee is directly attributable to Project A and represents an external direct cost of developing the software. Accordingly, the fixed subscription fee is capitalized. The fact that the fee is fixed and does not vary with usage does not, by itself, preclude capitalization. This conclusion differs from that in [Example 2](#) because the tool is dedicated to a single identified project and is only used for qualifying development activities, rather than being made broadly available throughout the organization.

Example 4

Entity C enters into an arrangement with a third-party AI provider to support both development and operational activities in its technology environment.

The key terms and facts associated with the arrangement are as follows:

- The pricing model includes a fixed monthly charge for compute capacity, together with variable usage-based charges (e.g., tokens and API calls).
- The AI coding agent is used by engineering teams to develop a new internal-use application, including code generation and testing activities during a period in which capitalization is appropriate.
- All token consumption associated with the software project can be tracked through project IDs and development environment tags.
- Vendor usage reports identify the number of tokens consumed by the project and employee along with the related activities associated with token-based charges.
- The fixed monthly capacity charge is not associated with any specific project and provides access to a shared platform used by multiple projects and activities at the organization.

Because the variable token-based charges are directly associated with a specific internal-use software project, can be attributed to qualifying development activities, and are supported by project-level usage data, C concludes that those usage costs represent direct external costs of services consumed in developing internal-use software. Accordingly, the token-based charges are capitalized as part of the cost of the internal-use software.

In contrast, the fixed monthly capacity charge provides access to shared infrastructure that supports multiple projects and activities at the organization. Entity C considered whether the usage data could be employed to directly attribute the fixed charge to the project but concluded that it could not, because the charge is incurred to maintain stand-ready capacity regardless of how much any particular project consumes. Because those costs cannot be directly attributed to a specific qualifying software development activity, C concludes that the fixed capacity charges are overhead costs and expenses them as incurred.

Example 5

Entity D uses a third-party AI coding agent priced on a per-token basis. An engineer initiates a single agent session that performs several activities in sequence.

The key terms and facts associated with the arrangement are as follows:

- Within one session, the software engineer has the agent generate and test code for Project A (a qualifying internal-use software project in its capitalization period) and investigates and helps remediate a production incident affecting an existing application (a nonqualifying activity).
- The AI service is priced by token consumption, and the vendor's usage logs record tokens consumed, time-stamped by prompt, tasks performed, and tokens consumed by the prompt.
- Entity D has developed detailed tracking processes and project management controls that link AI coding agent token usage to activity performance.
- Entity D can reliably associate specific token consumption with each activity by using the task-level logs, the repository identifiers, and the development ticket referenced in the session.

Because the session includes both qualifying and nonqualifying activities, D does not treat the entire session as a qualifying activity merely because part of it is related to Project A. Using the task-level usage logs supported by the tracking process, D identifies the tokens consumed in generating and testing code for Project A and capitalizes those charges as direct external costs of services consumed in developing internal-use software. The tokens consumed in investigating the production incident (production support) and drafting supporting documentation are expensed as incurred.

If the qualifying and nonqualifying token consumption had not been identified on a supportable basis, capitalization of the combined session cost would not have been appropriate. Rather, the session cost would have been expensed as incurred.

Example 6

Entity F enters into a one-year arrangement with an AI provider under which it commits to a minimum payment in exchange for a specified quantity of tokens.

The key terms and facts associated with the arrangement are as follows:

- Entity F commits to pay \$1 million for the right to consume up to 5 billion tokens during the year.
- Entity F expects to consume approximately 600 million tokens and does not expect to exceed the 5 billion included tokens or incur charges above the \$1 million minimum.
- The 5 billion token cap is deemed to be a protective measure and does not suggest that F is prepaying for a specified volume of tokens.
- The included tokens are available to multiple projects and activities at the organization, including new development, maintenance, research, and general productivity.
- Token consumption can be tracked by project and activity.

Because F will pay the same \$1 million regardless of whether it consumes 400 million, 600 million, or 900 million tokens, and because the 5 billion token cap is deemed to be protective, the payment functions economically as a fixed fee for access to AI services rather than as a prepayment for discrete units expected to be consumed. Dividing the fixed payment by tokens consumed and assigning that amount to projects on the basis of relative usage would merely represent an allocation of a fixed cost that F would incur regardless of any particular project's usage; it would not be a measurement of a cost directly attributable to a project.

Accordingly, F does not capitalize amounts assigned to projects on a per-token basis. If the minimum-commitment arrangement had instead been obtained specifically for an identified project and been usable only in qualifying development activities for that project, the minimum payment might have represented a direct project cost that was based on the project-specific nature of the contracted service rather than on an allocation to individual tokens consumed.

The arrangement in this example differs from the prepaid arrangement in [Example 7](#), in which the entity expects to consume substantially all of the purchased units and measures cost by units actually consumed.

Example 7

Entity G prepays for AI credits it expects to consume in full during the contract term.

The key terms and facts associated with the arrangement are as follows:

- Entity G prepays \$1 million for 1 million AI credits available over a 12-month period and expects to consume all of the credits.
- During a given month, 200,000 credits are consumed in qualifying development activities for Project A (in its capitalization period), 100,000 credits are consumed in maintaining existing software, and 50,000 credits are used for general research.
- Project-level usage records identify the projects and activities that consumed the credits.
- Entity G tracks token usage and can identify credits consumed in development, maintenance, and research activities.

Because G expects to consume the purchased units, the up-front payment represents a right to receive future AI services and is initially recorded as a prepayment. As credits are used, G reduces the prepaid balance and evaluates the related services on the basis of the activities they supported. The cost of the 200,000 credits consumed in qualifying development for Project A is capitalized. The credits consumed in maintenance and general research are expensed as the related services are received, and the cost of the remaining unconsumed credits continues to be reported as a prepaid amount, subject to consideration of expiration or other applicable accounting requirements.

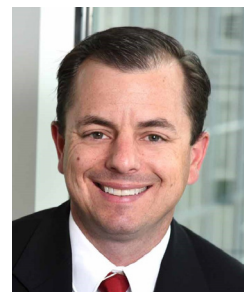
In this situation, G uses consumption records to measure the cost of services received as the prepaid balance is reduced; it is not allocating the up-front payment among projects. Accordingly, this arrangement is distinguished from the minimum commitment in [Example 6](#), in which the fixed payment would be incurred regardless of usage.

Contacts

If you have questions about this publication, please contact the following Deloitte industry professionals:



Aaron Shaw
U.S. Technology Industry
Professional Practice Director
Audit & Assurance
Partner
Deloitte & Touche LLP
+1 202 220 2122
aashaw@deloitte.com



Chris Chiriatti
U.S. Audit & Assurance
Managing Director
Deloitte & Touche LLP
+1 203 761 3039
cchiriatti@deloitte.com



Mark Strassler
U.S. Audit & Assurance
Managing Director
Deloitte & Touche LLP
+1 415 783 6120
mstrassler@deloitte.com



Hannah Higgins
U.S. Audit & Assurance
Senior Manager
Deloitte & Touche LLP
+1 617 960 8676
hhiggins@deloitte.com

For information about Deloitte's service offerings related to the accounting for internal-use software costs, please contact:



Chris Azar
U.S. Audit & Assurance
Partner
Deloitte & Touche LLP
+1 650 740 8745
cazar@deloitte.com



Dan Le
U.S. Audit & Assurance
Technology Sector Lead
Partner
Deloitte & Touche LLP
+1 206 716 6015
dle@deloitte.com



Jean-Denis Ncho Oguie
U.S. Audit & Assurance
TMT Industry Leader
Partner
Deloitte & Touche LLP
+1 415 783 6600
jnchooguie@deloitte.com

Dbriefs for Financial Executives

We invite you to participate in Deloitte's live [Dbriefs](#) webcasts for valuable insights into important developments affecting your business. The [Dbriefs for Financial Executives](#) series covers various accounting, governance, and financial reporting topics. Dbriefs also provide a convenient and flexible way to earn CPE credit — right at your desk.

Subscriptions

To subscribe to Dbriefs, or to receive accounting publications issued by Deloitte's Accounting and Reporting Services Department, please visit [My.Deloitte.com](#).

The Deloitte Accounting Research Tool

The [Deloitte Accounting Research Tool \(DART\)](#) is a comprehensive online library of accounting and financial disclosure literature. It contains material from the FASB, EITF, AICPA, PCAOB, and SEC, in addition to Deloitte's own accounting manuals and other interpretive guidance and publications.

Updated every business day, DART has an intuitive design and powerful search features that enable users to quickly locate information anytime, from any device and any browser. While much of the content on DART is available at no cost, subscribers have access to premium content, such as Deloitte's *FASB Accounting Standards Codification Manual*. DART subscribers and others can also [subscribe](#) to *Weekly Accounting Roundup*, which provides links to recent news articles, publications, and other additions to DART. For more information, or to sign up for a free 30-day trial of premium DART content, visit [dart.deloitte.com](#).

Industry Spotlight is prepared by members of Deloitte's National Office as developments warrant. This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor. Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

The services described herein are illustrative in nature and are intended to demonstrate our experience and capabilities in these areas; however, due to independence restrictions that may apply to audit clients (including affiliates) of Deloitte & Touche LLP, we may be unable to provide certain services based on individual facts and circumstances.

About Deloitte

Deloitte refers to one or more of Deloitte Touche Tohmatsu Limited, a UK private company limited by guarantee ("DTTL"), its network of member firms, and their related entities. DTTL and each of its member firms are legally separate and independent entities. DTTL (also referred to as "Deloitte Global") does not provide services to clients. In the United States, Deloitte refers to one or more of the US member firms of DTTL, their related entities that operate using the "Deloitte" name in the United States and their respective affiliates. Certain services may not be available to attest clients under the rules and regulations of public accounting. Please see [www.deloitte.com/about](#) to learn more about our global network of member firms.