KPMG

FASB issues ASU

Accounting for internal-use software costs

September 18, 2025 (updated February 17, 2026)

ASU 2025-06 modernizes the internal-use software guidance in Subtopic 350-40.

Source and applicability

- ASU 2025-06, Intangibles—Goodwill and Other—Internal-Use Software (Subtopic 350-40): Targeted Improvements to the Accounting for Internal-Use Software
- *All* entities that incur internal-use software costs, including website development costs

Fast facts, impacts, actions

The ASU is intended to modernize old internal-use software guidance written in 1998 to adapt to the agile (i.e. iterative and flexible) basis predominantly used to develop software today. For many entities it may also have the effect of more closely aligning the accounting for the development of software sold on a software-as-a-service (SaaS) basis with that of software licensed to customers. The *key* amendments do the following.

Change the cost capitalization threshold by:

- eliminating accounting consideration of software project development stages; cost capitalization would begin when (1) management has authorized and committed to funding the project and (2) it is 'probable' the project will be completed and the software used to perform its intended function (the 'probable-to-complete' threshold); and
- enhancing the guidance around the 'probable-to-complete' threshold (given its new prominence) and providing new examples in Subtopic 350-40 to illustrate its application.

Modify the website development costs guidance by eliminating Subtopic 350-50 and relocating any remaining relevant guidance into Subtopic 350-40 and adding a new example.

The ASU does not change: (1) the existing accounting requirements for external-use software (i.e. software to be sold or licensed) development costs, (2) *what* internal-use software costs can be capitalized (e.g. data conversion/migration, training and software maintenance costs would continue to be expensed as incurred), or (3) when internal-use software cost capitalization ceases (i.e. when the software is 'substantially complete and ready for its intended use').

Background

By amending Subtopic 350-40, the ASU responds to stakeholder feedback, primarily from preparers and practitioners, that indicated the FASB should focus on modernizing the accounting for software costs. Stakeholders told the FASB the existing guidance is outdated for two reasons.

- It mostly presumes a 'waterfall' (i.e. sequential and linear) method of software development, which was predominant when the existing guidance was developed in the late 1990s. However, the agile method for developing software that is predominant today does not fit well with the existing guidance.
- Having different cost accounting models for software that is licensed to customers versus software that is sold to customers only on a SaaS basis can lead to different accounting outcomes for economically similar transactions.

Additionally, investors indicated a need for greater transparency around entities' software costs.

In June 2022, the FASB added a software cost project to its technical agenda. Between then and March 2024, the FASB considered several *larger* changes to entities' accounting for internal- *and* external-use software costs. Those changes included adopting a *single* software cost accounting model for those two types of software. However, no consensus developed around what those larger changes should be, including whether internal- and external-use software should be governed by a single accounting model. Therefore, the FASB ultimately decided to solely focus on more limited amendments to Subtopic 350-40, and to leave Subtopic 985-20 on external-use software unchanged.

Similarly, while more significant new disclosure requirements were considered (e.g. a required annual rollforward of entities' capitalized software costs), the final ASU only, in effect, clarifies that entities are required to provide Subtopic 360-10 disclosures for *all* capitalized internal-use software.

Changes to the cost capitalization threshold

Removal of project development stages

A principal objective of ASU 2025-06 is to modernize the dated Subtopic 350-40 guidance for pervasive changes in how entities develop software. The waterfall method contemplated several discrete development stages (i.e. preliminary project, application development and postimplementation-operation). Development moved methodically from one stage to the next only when the one preceding it was completed.

By contrast, now most software development is 'agile'. Agile software development is flexible and iterative. That is, while there is an overall development objective, agile projects are generally much more lightly planned and completed through a series of shorter time-frame development 'sprints'. It is understood and accepted that later sprints will frequently drive re-work or revision of tasks completed in previous sprints to arrive at the completed project. This intended ability to make changes (flexibility) and revisit and/or reperform earlier activities (iteration) often gives rise to the sense that discrete (or distinct) software development stages – i.e. that each begin only after the one preceding it ends – do not exist in agile projects. Agile and waterfall software development are discussed in further detail in sections 3.2.60 and 3A.2.10 of KPMG Handbook, Software and website costs.

While practice has developed around applying Subtopic 350-40 to agile software development projects, such practice has not always been consistent. Additionally, the incongruence of the existing guidance with the predominant method of software development creates complexity. Question 3.2.170 and section 3A.2.10 in KPMG Handbook, Software and website costs, discuss these complexities and relevant considerations.

To better align the guidance with the predominantly agile nature of current software development, ASU 2025-06's amendments eliminate all references to project stages in Subtopic 350-40. This leaves all entities, regardless of software development method (i.e. agile, waterfall, a hybrid of those, or otherwise), to rely solely on the *remaining* criteria in paragraph 350-40-25-12 to determine when development cost capitalization should begin. Those are whether (1) management has authorized and committed to funding the software project and (2) it is probable that the project will be completed and the software used to perform the function(s) intended (the 'probable-to-complete threshold'). Until (1) and (2) are met, all software development costs are expensed as incurred.

Removing the existing staging guidance seems likely to substantially alleviate the challenges that exist in applying the current guidance arising from entities' use of an agile (or similar non-linear, non-sequential) software development method. However, as discussed below, some of the judgments eliminated may be replaced with new judgments around the 'probable-to-complete' threshold.

Defining a software project

The Board considered but decided not to define 'software project' or provide specific guidance on what constitutes a software project. It observed that this unit of account has never been defined and doing so now might change practice or limit the exercise of reasonable judgment, neither of which was an objective of the ASU.

Defining a 'software project' involves judgment. Generally, we believe a software project is defined by its functional independence from other development efforts (e.g. whether the successful release or deployment of the software depends on the success of other ongoing efforts). Section 3A.2.20 in KPMG Handbook, Software and website costs, addresses identifying 'software projects' further.

Definition of 'probable'

The ASU's amendments explicitly link the term 'probable' used in the probable-to-complete threshold to the ASC Master Glossary definition. 'Probable' is defined in the Master Glossary as "The future event or events are likely to occur."

AICPA Statement of Position (SOP) No. 98-1 (the source of the Subtopic 350-40 guidance) stated that 'probable' had the same meaning therein as it had in FASB Concepts Statement No. 6 (CON 6), where it was defined and used in its conventional sense (in short, that which can be reasonably expected). However, Subtopic 350-40 never incorporated this statement from SOP 98-1 and the CON 6 probable definition was eliminated when CON 6 was superseded. Therefore, differing views have arisen about which probable definition applies – i.e. that originally intended or the existing Master Glossary definition.

The Board decided to explicitly link 'probable' in Subtopic 350-40 to the Master Glossary definition to clarify which probable definition applies and because the Master Glossary definition is generally well understood and widely used elsewhere in US GAAP. By contrast, CON 6 no longer exists.

Probable-to-complete threshold changes

The ASU amends the existing probable-to-complete threshold by adding considerations to the evaluation and providing new examples about its application.

While the threshold itself remains unchanged, the ASU adds guidance to state that an entity does not meet this threshold if there is 'significant uncertainty' as to the software's development. Significant development uncertainty is assessed for each software project and is considered to exist if either of the following factors is present.

Novel, unique, unproven functions and features or technological innovations	The software has novel, unique, unproven functions and features or technological innovations that have not yet been proven through coding and testing. Subtopic 985-20 has long required entities to (1) consider the existence and resolution of 'high-risk development issues', defined consistent with the above, when assessing the 'technological feasibility' of external-use software, and (2) resolve such issues through coding and testing. Therefore, this new Subtopic 350-40 assessment and resolution may be similar in some respects. Observations below provide additional thoughts on this significant development uncertainty factor.
Software's significant 'performance requirements'	The amendments require entities to determine whether the software's *significant* performance requirements have been determined and are no longer subject to *substantial* revision. They also define software's 'performance requirements' as "what an entity needs the software to do (for example, functions or features)."

The amendments explicitly state that it may require less judgment to determine whether significant development uncertainty exists for some software projects (e.g. to customize and implement established third-party software). Therefore, as a practical matter, only limited effort may be necessary.

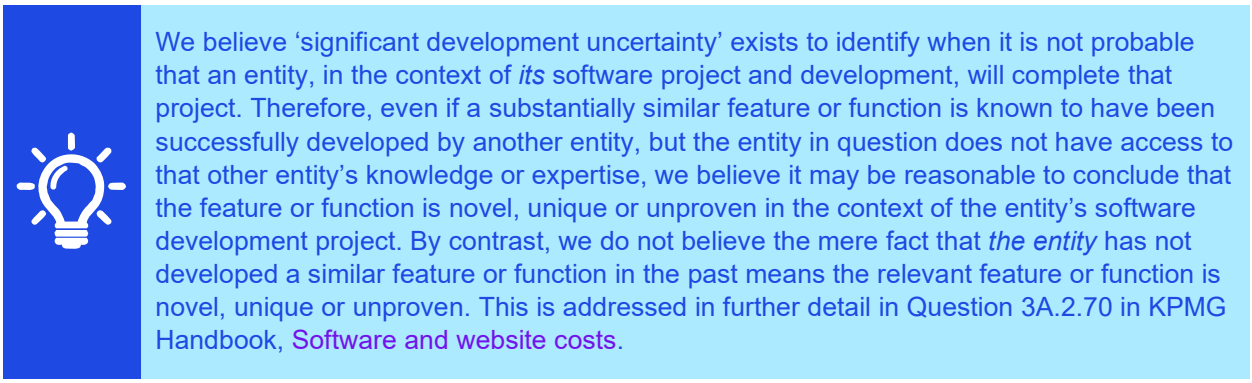
The Board intentionally did not create a finite list of factors or indicators related to meeting the overall probable-to-complete threshold (noting that this threshold criterion already exists in Subtopic 350-40). Entities will continue to consider all relevant facts and circumstances when assessing the probable-to-complete threshold, and not just the existence (or non-existence) of significant development uncertainty.

Significant development uncertainty observations

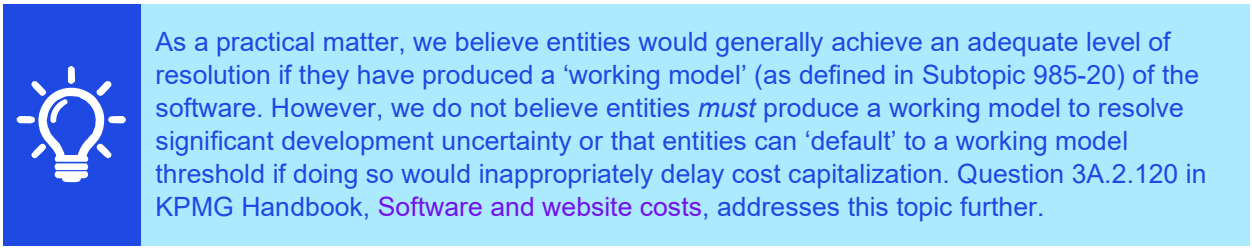
The following reflect considerations we believe are important for entities to contemplate as they move toward implementing the amended guidance.

Novel, unique or unproven functions or features

The ASU does not specify whether an entity must assess whether a function or feature is novel, unique or unproven (or reflects a technological innovation) based on its own experience and expertise (e.g. whether *it* has successfully developed a function or feature of this nature before) or should, instead, assess this against whether a similar function or feature exists more broadly. Therefore, reasonable judgment may be acceptable in this regard.

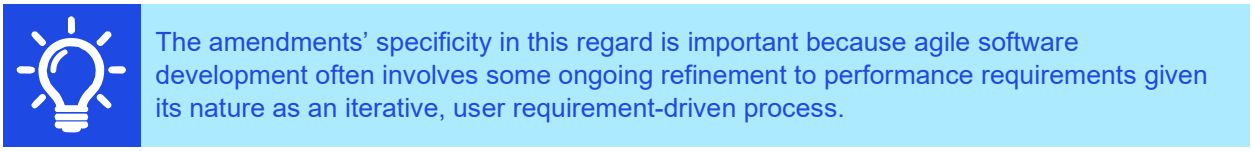


We believe the requirement to resolve significant development uncertainty stemming from novel, unique or unproven features or functions through ‘coding and testing’ practically refers to sufficiently establishing the viability of those features or functions such that they are not an impediment to the successful completion of the software project.



Significant performance requirements

The FASB emphasizes in the basis for conclusions to the ASU that the amendments do not require an entity to identify and resolve all of the software’s performance requirements before it begins to capitalize software development costs, only those performance requirements that are ‘significant’ and *substantively* unresolved. Therefore, an entity should not defer eligible cost capitalization for either (1) minor performance requirements that have not yet been determined or (2) significant performance requirements subject only to further *minor* revision.



Judgments

The FASB acknowledged that evaluating whether the probable-to-complete threshold has been met requires judgment, including when assessing whether significant development uncertainty exists. An entity’s judgment should be based on its evaluation of its specific facts and circumstances. The Board noted that the application of judgment is inherent in US GAAP and concluded that the use of judgment is appropriate in the internal-use software guidance because an entity’s management is best positioned to evaluate its facts and circumstances, considering the diverse and continuously evolving nature of software development.

Entities may make different judgments, even in similar circumstances, about whether software features or functions are novel, unproven or unique; what constitutes a 'significant' performance requirement; or what level of ongoing revision of a significant performance requirement is 'substantial' (see Question 3A.2.60 in KPMG Handbook, Software and website costs). They also may interpret the level of coding and testing necessary to resolve significant development uncertainty stemming from novel, unproven or unique features and functions differently.

SaaS software development

The FASB states in the final ASU that the probable-to-complete threshold changes could bring closer alignment between the accounting for external-use software development costs – the vast majority of which are expensed as incurred by most entities – and similar costs to develop software that will be sold by an entity only on a SaaS basis (SaaS software). In particular, the additional considerations around significant development uncertainty may result in entities expensing significant portions of their SaaS software development costs if they conclude that significant development uncertainty exists – and therefore the software is not probable of completion – until relatively late in the development process.

Software embedded in a tangible product

The proposed ASU would have created guidance on whether a tangible asset with embedded software is a single unit of account (inclusive of the embedded software) or two separate units of accounting (i.e. the tangible asset accounted for separately from the embedded software). The *final* ASU does not include such guidance.

We believe practice already exists to typically treat any embedded software that is integral (or essential) to the functionality of a tangible asset as simply *part of* that tangible asset (i.e. not as a separate unit of account). See Question 2.4.110 in in KPMG Handbook, Software and website costs.

Entities selling tangible products with embedded software ('firmware') apply the external-use software guidance in Subtopic 985-20 to firmware development.

Financial statement presentation and disclosure

Cash flow presentation of internal-use software costs

The proposed ASU would have required cash paid for capitalized internal-use software costs to be presented separately from other investing cash outflows in the statement of cash flows. However, in a split decision, the Board voted to not carry forward this requirement to the final ASU based on stakeholder feedback.

Cash flow presentation of capitalized cloud computing arrangement implementation costs

The Board considered but decided not to change the existing requirement in paragraph 350-40-45-3 that these costs are required to be presented as operating outflows in the statement of cash flows (see Question 7.2.50 in KPMG Handbook, Software and website costs).

Disclosures

Subtopic 350-40 will now expressly state that entities must provide the disclosures required under Subtopic 360-10 on PP&E to capitalized internal-use software and related amortization, regardless of how the internal-use software is classified on the balance sheet (e.g. as PP&E or an intangible asset) or how it was acquired (e.g. internally developed or licensed from a third party). This may be an important

clarification given that *licensed* internal-use software is required to be classified as an acquired intangible asset.

For internal-use software classified as an intangible asset, we believe entities may need to consider providing at least certain of the now-required Subtopic 360-10 disclosures *separately* from those it provides for assets actually classified as PP&E to prevent them from being confusing or misleading to users of the financial statements (e.g. so that total depreciation amounts for the period reconcile to accumulated depreciation changes). This may also include providing these disclosures in the intangible assets note, where users may reasonably expect to find disclosures about those assets presented as intangible assets on the entity's balance sheet.

Website development costs and other Codification amendments

Subtopic 350-50 (website development costs) heavily leverages the guidance in Subtopic 350-40, including a requirement to expense or capitalize based on the development stage of the website project. The ASU eliminates Subtopic 350-50 and incorporates key, non-development stage website-specific development costs guidance into Subtopic 350-40. It also adds a new example illustrating the application of Subtopic 350-40 to website development. Other Codification amendments were also made to be consistent with the ASU's amendments to Subtopic 350-40 (e.g. to Subtopic 720-45 on business process re-engineering).

Modern website development is often similar in nature and scope to software application development. Therefore, the Board decided that separate website development cost guidance was no longer useful or needed. However, because Subtopic 350-50 has limited guidance and Subtopic 350-40 will now include the important elements from Subtopic 350-50, we do not believe the amendments will significantly change practice.

Effective dates

Effective dates	All entities
Annual and interim periods – Fiscal years beginning after…	December 15, 2027
Early adoption permitted?	Yes, in any interim or annual period for which an entity's financial statements have not been issued (or made available for issuance) as of the beginning of the entity's fiscal year.

Transition

The ASU requires entities to adopt the new Subtopic either (1) retrospectively, (2) prospectively to software costs incurred after the adoption date (i.e. on existing, in-process software projects or new projects) or (3) on a modified prospective basis.

- The retrospective approach will result in a cumulative effect adjustment to retained earnings (or other appropriate components of equity or net assets) as of the beginning of the first year presented in the entity's first set of annual financial statements issued after adoption.

- The prospective transition option is generally consistent with that offered to entities when adopting ASU 2018-15 on cloud computing arrangement implementation costs.
- The modified prospective approach requires:
  - prospective application of the amendments to software costs incurred on new projects and software projects in process as of the adoption date ('in-process projects'); and
  - a cumulative effect adjustment through retained earnings (or other appropriate caption) for any in-process project capitalized costs that do not qualify for capitalization under the amended guidance as of the adoption date.

Chapter 9 in KPMG Handbook, Software and website costs, addresses the effective dates and transition requirements of ASU 2025-06, including interpretive guidance and examples.

Transition disclosures

Entities are required to provide appropriate transition disclosures under Topic 250; specific requirements differ depending on the transition method chosen.

Contributing authors
Scott Muir, P.K. Barot, Lori Kozen

KPMG Financial Reporting View
kpmg.com/us/frv

This newsletter is part of our Defining Issues® collection of newsletters and articles with insights and news about financial reporting and regulatory developments.

Sign up here to receive news and insights delivered to your mailbox.