



BLUEPRINT: A BDO SERIES

Accounting for Software Costs

July 2026



TABLE OF CONTENTS

Introduction	4
Accounting for Software Costs: Scope	4
Software to be Sold, Leased, or Marketed	5
Internal-Use Software and Implementation Costs of CCA	6
Presentation and Disclosure	6
About This Publication	7
Acknowledgements	8
Chapter 1: Scope	9
1.1 Overview	9
1.2 Software to Be Sold, Leased, or Marketed	10
1.3 Internal-Use Software	12
1.4 Hosting Arrangement or CCA	14
1.5 Software Acquired in a Business Combination	15
1.6 Software in Research and Development Activities	15
1.7 Software Costs in Contracts With Customers	17
Chapter 2: Software to Be Sold, Leased, or Marketed	19
2.1 Overview	19
2.2 Establishing Technological Feasibility	20
2.3 Capitalizable costs	24
2.4 Subsequent Accounting for Capitalized Software Costs	29
2.5 Product Enhancements	31
2.6 Issues Arising After Establishing Technological Feasibility	33
2.7 Modifications to Features Originally Planned for the Product	33
Chapter 3: Internal-Use Software and Implementation Costs of CCA	34
3.1 Overview	34
3.2 Capitalization Criteria and Capitalizable Costs	35
3.3 Subsequent Measurement	44
3.4 Upgrades and Enhancements	48
3.5 Implementation Costs of CCA that is a service contract	49
3.6 Website Development Costs	52
Chapter 4: Presentation and Disclosures	55
4.1 Overview	55
4.2 Presentation	55
4.3 Disclosures	58

Chapter 5: Transition	61
5.1 Overview	61
5.2 Effective Date	61
5.3 Transition Approaches	62
5.4 Transition Disclosures	64
Appendix A: Other BDO Publications	65
Contacts	66



Introduction

Software is a set of codes or programs that tells hardware, for example, computer servers, laptops, phones and other devices, what to do. Entities across various industries develop or acquire software for internal use including supporting internal operations and enabling the provision of goods or services to end users such as online banking, online shopping, and online games. Entities in the technology industry generally sell software to their end users through either:

- ▶ On-premise licenses allowing an end user to use the software offline
- ▶ Arrangements where an end user can **only** access and use the software through the internet, for example, in a software-as-a-service (SaaS) arrangement.

Additionally, entities often embed software into tangible items, for example, network equipment, cars, electronic appliances, and industrial robots. As a result, every entity uses software to some extent and therefore incurs software costs and is affected by the accounting guidance for software costs.

U.S. GAAP includes multiple models to account for the costs of developing, acquiring, and implementing software. The applicability of a particular model depends on how an entity acquires or develops software and how the entity intends to use that software. Because of the judgments inherent in the guidance, entities may face challenges in determining which guidance applies to their software costs and applying that guidance.

Additionally, continuous evolution of the software industry and technological advancement affect how software products are developed and sold, creating complexities in applying the guidance on software costs. Also, as artificial intelligence (AI) training and data conversion costs become more prevalent, new practice issues may arise that are not directly addressed by the guidance.

This publication discusses the framework in U.S. GAAP for accounting for software costs and related practice issues, including accounting for costs incurred to develop or acquire software for external use as well as internal use.

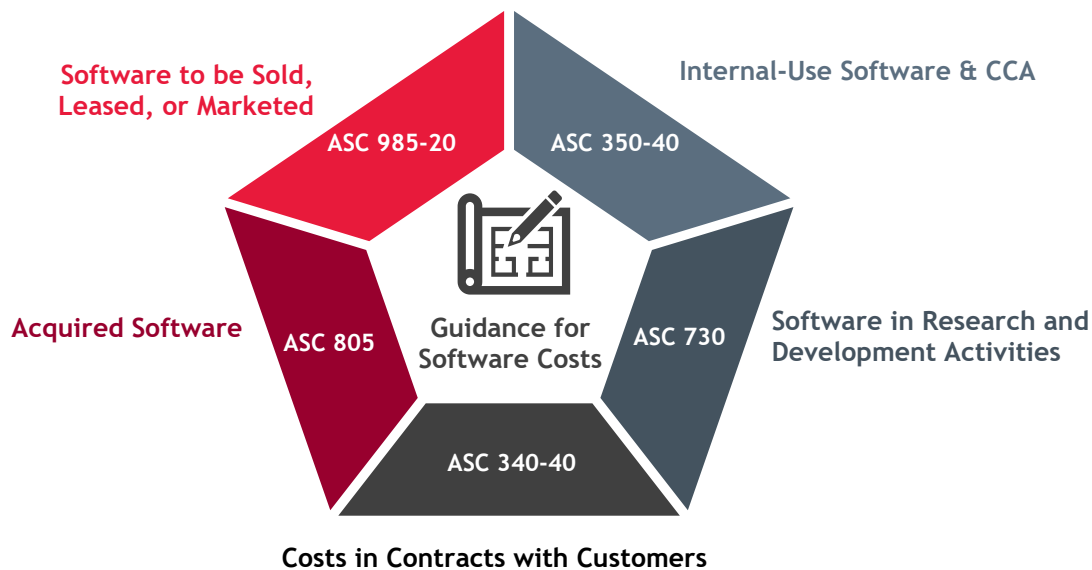
This publication incorporates the guidance in Accounting Standards Update (ASU 2025-06), *Intangibles – Goodwill and Other – Internal-Use Software (Subtopic 350-40): Targeted Improvements to The Accounting For Internal-Use Software*, which was issued in September 2025 to modernize the internal-use software guidance to accommodate the evolution in software development method from a linear development method (for example, waterfall) to a nonlinear development method (for example, agile).

ACCOUNTING FOR SOFTWARE COSTS: SCOPE

Accounting for software costs depends on how an entity intends to use the resulting software. ASC 350-40, *Intangibles – Goodwill and Other – Internal-Use Software*, provides a model for internal-use software while ASC 985-20, *Software – Costs of Software to Be Sold, Leased, or Marketed*, provides a model for software intended for sale, lease, or marketing. The internal-use software guidance in ASC 350-40 also addresses situations when an entity provides access to the software through a hosting arrangement or cloud computing arrangements (CCA), for example, SaaS. For software outside the scope of ASC 350-40 or ASC 985-20, other U.S. GAAP may apply.

There are significant differences between the guidance on accounting for software costs under ASC 350-40, ASC 985-20, and other U.S. GAAP. Further, the guidance in ASC 350-40 that applies to software licensing arrangements differs in important respects from the guidance in ASC 350-40 that applies to CCAs. It is important to correctly identify the applicable guidance to appropriately account for the costs of developing or acquiring software because of the significant differences therein.

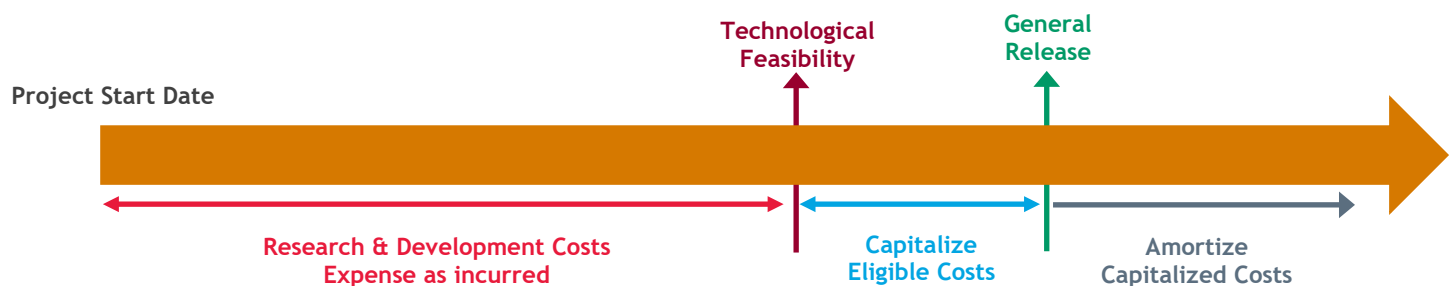
The following diagram shows the accounting models applicable to software costs:



SOFTWARE TO BE SOLD, LEASED, OR MARKETING

ASC 985-20 provides guidance for accounting and reporting for the costs of computer software to be sold, leased, or otherwise marketed as a separate product or as part of a product or process (externally marketed software), whether internally developed and produced or purchased. ASC 985-20 specifies whether an entity must expense or capitalize the software development costs based on whether the costs are incurred before or after the product reaches technological feasibility:

- ▶ Costs incurred before reaching technological feasibility are considered research and development costs and are expensed when incurred.
- ▶ Costs to develop, produce or purchase externally marketed software are capitalized after technological feasibility is reached. Capitalization of eligible costs ceases once the product is available for general release.



Additionally, ASC 985-20 provides guidance on subsequent accounting for capitalized costs and accounting for costs incurred after the product is available for general release and product enhancements.

INTERNAL-USE SOFTWARE AND IMPLEMENTATION COSTS OF CCA

ASC 350-40 provides guidance for accounting and reporting for the costs of internal-use software, including examples that illustrate whether computer software is in scope or not. ASC 350-40 requires an entity to begin capitalizing internal-use software development costs when **both** of the following occur:



Management, with the relevant authority, implicitly or explicitly **authorizes and commits** funding for the software project.

Examples of authorization and commitment to funding a software project include the execution of a contract with a third party to develop the software, approval of expenditures related to internal development or a commitment to obtain the software from a third party.



It is **probable** the project will be completed, and the software will be used to perform the function intended (referred to as “**probable-to-complete recognition threshold**”).

In evaluating whether the probable-to-complete recognition threshold has been met, an entity assesses whether there is significant uncertainty associated with the development activities of the software, referred to as “**significant development uncertainty.**” (see Section 3.2.1.1).

If the capitalization criteria are no longer met for software being developed, an entity cannot capitalize further costs and must apply the guidance in Section 3.3.1 on impairment to existing asset balances.

An entity must **cease** capitalizing internal-use software costs no later than the point at which a software project is substantially complete and ready for its intended use, that is, after all substantial testing is complete.

Subsequently, an entity amortizes the costs of software developed or obtained for internal use on a straight-line basis unless another systematic and rational basis is more representative of the software's use. Amortization begins when the software is ready for its intended use (that is, after all substantial testing is complete), regardless of when the software is placed in service. An entity must also assess internal-use software for impairment in accordance with ASC 360-10-35, which applies to property, plant, and equipment.

This guidance also applies to website development costs, as well as to costs to implement a hosting arrangement or CCA that is a service contract (for example, SaaS).

PRESENTATION AND DISCLOSURE

ASC 350-40 and ASC 985-20 provide limited guidance on presentation and disclosure requirements for software costs related to software to be sold, leased, or marketed, internal-use software, and cloud computing arrangements.

Presentation guidance addresses the classification of capitalized software costs on the balance sheet, the presentation of amortization in the income statement, and the categorization of related cash flows, with additional considerations for purchased software and implementation costs in cloud computing arrangements (see Sections 4.2, 4.2.1, and 4.2.2).

Disclosure requirements include unamortized software costs and amounts expensed. Recent amendments under ASU 2025-06 require entities to align their internal-use software disclosures with those for property, plant, and equipment (see Sections 4.3, 4.3.1, and 4.3.2).

If ASC 350-40 or ASC 985-20 does not provide specific presentation or disclosure requirements, an entity applies the general financial statement presentation principles and other U.S. GAAP, as appropriate.

ABOUT THIS PUBLICATION

This publication reflects the key aspects of the guidance on accounting for software development costs from:

- ▶ ASC 985-20, *Costs of Software to Be Sold, Leased, or Marketed*
- ▶ ASC 350-40, *Internal-Use Software* as amended by the following ASUs (collectively referred to as “ASC 350-40”):
 - ASU 2015-05, *Intangibles – Goodwill and Other – Internal-Use Software (Subtopic 350-40): Customer’s Accounting for Fees Paid in A Cloud Computing Arrangement*
 - ASU 2016-19, *Technical Corrections and Improvements*
 - ASU 2018-15, *Intangibles – Goodwill and Other—Internal-Use Software (Subtopic 350-40): Customer’s Accounting for Implementation Costs Incurred in A Cloud Computing Arrangement That Is a Service Contract*
 - ASU 2025-06, *Intangibles – Goodwill and Other—Internal-Use Software (Subtopic 350-40): Targeted Improvements to The Accounting for Internal-Use Software*



FASB UPDATE: AMENDMENTS TO THE INTERNAL-USE SOFTWARE GUIDANCE IN ASC 350-40

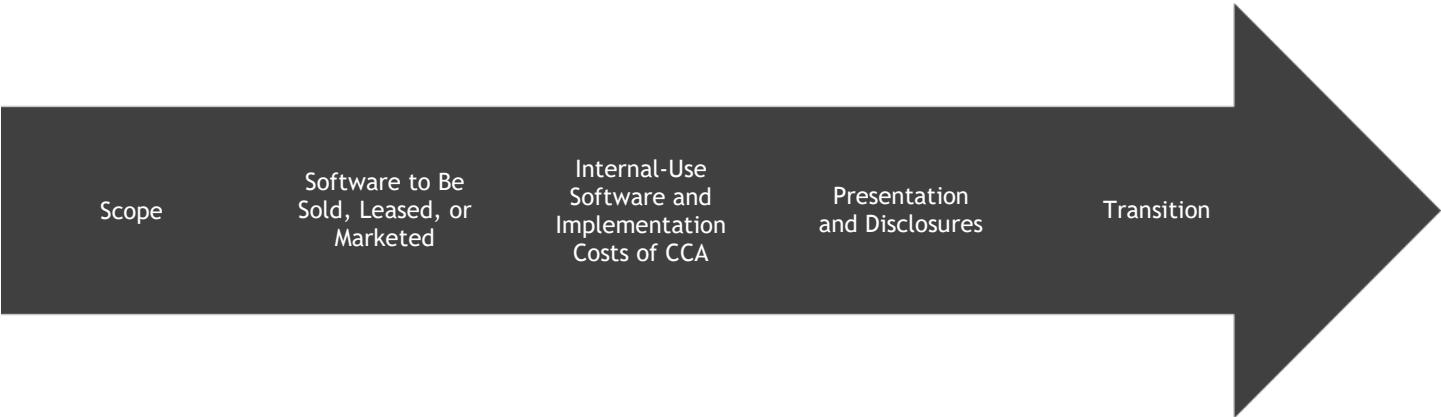
The FASB issued ASU 2025-06 in September 2025 to amend the internal-use software guidance in ASC 350-40. The amendments modernize the internal-use software guidance to accommodate the evolution in software development method from a linear development method (for example, waterfall) to a nonlinear development method (for example, agile).

Additionally, ASU 2025-06 supersedes the previous guidance for website development costs in ASC 350-50, *Website Development Costs*, and incorporates the recognition requirements for website-specific development costs into ASC 350-40, aligning it with the new guidance for internal-use software.

This publication assumes that an entity has adopted ASU 2025-06. See Chapter 5 for transition guidance.

This publication includes detailed guidance and diagrams on analyzing and accounting for software costs under U.S. GAAP. While it does not address all requirements under U.S. GAAP, it summarizes key concepts that are commonly considered in accounting for software costs. It also includes practical examples and interpretive guidance to assist entities and practitioners in accounting for software costs. The diagrams are provided to assist readers in understanding various aspects of the guidance on software costs. Accounting for software costs varies based on the specific facts and circumstances and, therefore, may differ from the examples and insights in this publication.

This publication is divided into following chapters that address key aspects of the guidance in U.S. GAAP to account for software costs:



ACKNOWLEDGEMENTS

This Blueprint, published by BDO's Professional Practice, is the culmination of efforts of many individuals, to whom we express our sincere appreciation.

- ▶ Bobbi Gwinn
- ▶ Adam Brown
- ▶ Angela Newell
- ▶ Hank Galligan
- ▶ Meredith Taylor
- ▶ Smruthi M N

Chapter 1: Scope

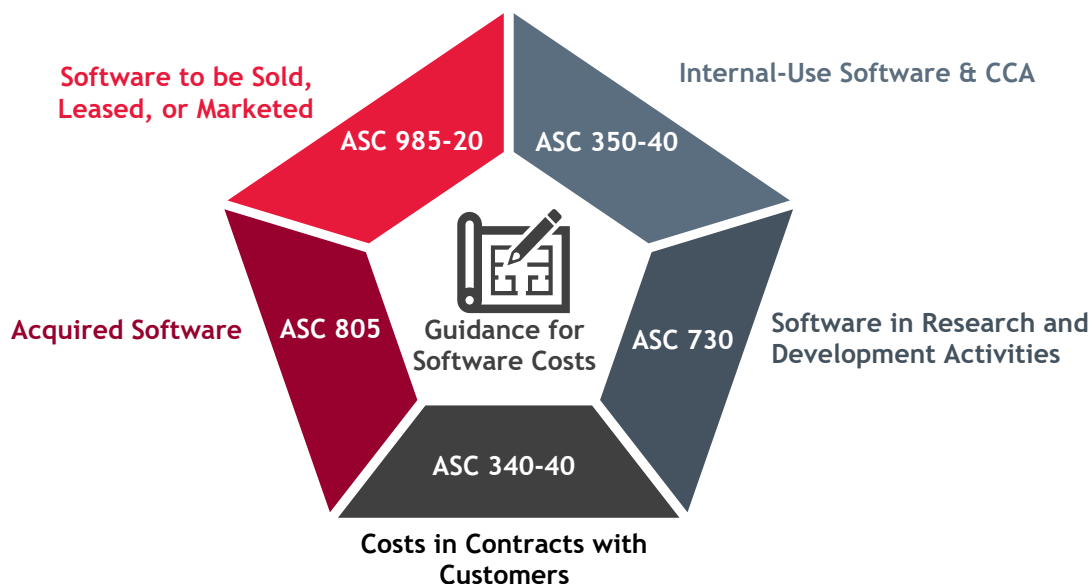


1.1 OVERVIEW

Accounting for software costs depends on how an entity intends to use the resulting software. ASC 350-40 provides a model for internal-use software while ASC 985-20 provides a model for software intended for sale, lease, or marketing. The internal-use software guidance in ASC 350-40 also addresses situations when an entity provides access to the software through a hosting arrangement or cloud computing arrangements (CCA), for example, SaaS. For software outside the scope of ASC 350-40 or ASC 985-20, other U.S. GAAP may apply. For example, ASC 340-40, *Other Assets and Deferred Costs – Contracts with Customers*, applies to software costs incurred to fulfill contracts with customers and ASC 730, *Research and Development*, applies to software costs related to research and development activities. Additionally, ASC 805, *Business Combinations*, applies to certain acquired software.

There are significant differences between the guidance on accounting for software costs under ASC 350-40, ASC 985-20 and other U.S. GAAP including, ASC 340-40, ASC 730, and ASC 805. Further, the guidance in ASC 350-40 that applies to software licensing arrangements differs in important respects from the guidance in ASC 350-40 that applies to CCAs. It is important to correctly identify the applicable guidance to appropriately account for the costs of developing or acquiring software because of the significant differences therein.

The following diagram shows the guidance on software costs:



BDO INSIGHTS: DETERMINING APPROPRIATE SCOPE

An entity's accounting for software costs is affected by how it acquires or develops and uses software. Given the complexity and evolution in how a software product is developed and sold, an entity may need to apply judgment based on the facts and circumstances to determine the appropriate guidance to apply.

**FASB UPDATE: AMENDMENTS TO THE INTERNAL-USE SOFTWARE GUIDANCE IN ASC 350-40**

The FASB issued ASU 2025-06 in September 2025 to amend the internal-use software guidance in ASC 350-40. The amendments modernize the internal-use software guidance to accommodate the evolution in software development methods from a linear development method (for example, waterfall) to a nonlinear development method (for example, agile). ASU 2025-06 removes references to project development phases and provides new recognition guidance to determine whether and when an entity must capitalize internal-use software costs.

This publication assumes that an entity has adopted ASU 2025-06. See Chapter 5 for transition guidance.

1.2 SOFTWARE TO BE SOLD, LEASED, OR MARKETED

**FASB REFERENCES**

ASC 985-20-15-1 through 15-5, ASC 985-20-55-1 through 55-3, and ASC 350-40-15-4A

ASC 985-20 provides guidance on accounting and reporting for the costs of software to be sold, leased, or otherwise marketed as a separate product or part of a product or process, whether internally developed or purchased. For example, ASC 985-20 applies to software sold as an on-premise software license and software embedded in a chip that is a part of a glucose monitor sold to a customer. Software embedded in a product and sold as part of the product is referred to as “firmware.” Common examples include operating systems in smart phones and robots.

Software used by an entity to produce a good or provide a service is not considered externally marketed software unless that software is included in or part of the good or service sold. However, ASC 985-20 applies to software used in providing some services to customers that would not be possible without the software when the software is transferred to the customer. For example, software that is installed on a customer's servers and used by an entity to monitor, collect, and analyze the customer's data would be subject to ASC 985-20.

ASC 985-20 **does not** apply to the following transactions and activities:

Internal-Use Software	Software developed or obtained for internal use which are accounted for under ASC 350-40. This includes software accessed through a hosting arrangement such as SaaS (see Section 1.3 and Chapter 3).
Acquired Research and Development Assets	Research and development assets acquired in a business combination, acquired in an acquisition by a not-for-profit entity, or recognized by a joint venture upon formation. If tangible and intangible assets acquired in those transactions are used in research and development activities, they are recognized and measured at fair value under ASC 805-20 (see BDO's Blueprint, Business Combinations Under ASC 805).
Costs in a Contract with a Customer	Arrangements to deliver software or a software system (either alone or together with other products or services) that require significant production, modification, or customization of software are accounted for as costs to fulfill a contract under ASC 340-40 (see Section 7.7 of BDO's Blueprint, Revenue Recognition Under ASC 606).

1.2.1 Interpretation of the Term “Software Product”

The terms “computer software product,” “software product,” and “product” encompass a computer software program, a group of programs, and a product enhancement. A software product is most easily defined by describing its necessary qualities: as a product, it is complete and has exchange value; as software, it is a set of programs that interact with each other. A program is defined as a series of instructions or statements that cause a computer to do work.

1.2.2. Software That is Both Licensed and Provided as a SaaS Arrangement

In some instances, a software vendor may plan to offer its software to customers both under licensing arrangements and as a SaaS arrangement. In those instances, the entity must determine which accounting literature to apply, ASC 350-40 or ASC 985-20. As described in Section 1.3.1, ASC 350-40-15-2A states that if an entity has developed a substantive plan to market the software as a license, the related costs are outside the scope of ASC 350-40. Therefore, if an entity has developed a substantive plan to market the software as a license, the entity must apply ASC 985-20 to the related development costs. If instead the entity has not developed a substantive plan to license the software product, it accounts for the development costs under ASC 350-40.

1.2.3 Hybrid Cloud Arrangements

An entity may deliver a software solution that includes both functionality provided via an on-premise or on-device software license and functionality delivered via a cloud-based SaaS solution. For example, an on-premise license may allow the customer to collate and analyze its internal data while the related cloud-based service may allow the customer to compare that data to market data. This type of arrangement is referred to as a hybrid cloud arrangement.

1.2.3.1 Vendor Accounting

The costs associated with developing the on-premise or on-device software license are in the scope of ASC 985-20 because the customer takes control of the license. However, the costs associated with developing cloud-based functionality are in the scope of ASC 350-40 because the entity uses that software to provide a service to the customer.

BDO INSIGHTS: UNIT OF ACCOUNT FOR HYBRID CLOUD ARRANGEMENTS

As noted above, costs to develop software to be sold and marketed are accounted for separately from costs to develop internal-use software in a hybrid cloud arrangement. In some situations, the vendor may conclude that the hybrid arrangement represents a single performance obligation under ASC 606, *Revenue from Contracts with Customers*. However, ASC 985-20 and ASC 350-40 do not include guidance that links the unit of account to the guidance on performance obligations in ASC 606. Therefore, we believe that the vendor's conclusions under ASC 606

are not relevant for determining whether to account for the costs associated with developing the on-premise license separately from the costs associated with developing cloud-based functionality.

However, in some instances the on-premise or on-device software may serve primarily as an interface to the cloud-based functionality rather than performing significant functionality itself. Those types of arrangements are often called “thin-client applications.” If the on-premise or on-device component of the hybrid cloud arrangement does not provide sufficient functionality and only enables connection to the cloud-based functionality, we believe it does not meet the definition of externally marketed software, and thus any related costs would not be accounted for separately under ASC 985-20.

Determining the unit of account in a hybrid cloud arrangement requires the application of professional judgment based on the facts and circumstances.

1.2.3.2 Customer Accounting

Because the customer is purchasing the on-premise or on-device license, the arrangement is in the scope of ASC 350-40. In addition, any costs of implementing the cloud-based SaaS services are also accounted for using the guidance in ASC 350-40 (see Section 3.5). However, the presentation of capitalized costs associated with hosting arrangements and the related amortization are treated differently than those related to capitalized license costs. Therefore, a customer must separately account for the two components of the hybrid arrangement. ASC 350-40 provides guidance on allocating the cost of internal-use software to one or more elements in a multi-element arrangement (see Section 3.2.4).

1.3 INTERNAL-USE SOFTWARE



FASB REFERENCES

ASC 350-40-15-1 through 15-4 and ASC 350-40-55-1 through 55-2

ASC 350-40 provides guidance on accounting and reporting for the costs of software developed or obtained for internal use. ASC 350-40 specifically includes and excludes the following transactions and activities:

Scope Inclusions	Scope Exclusions
▶ Software used in an entity’s operations, including software provided to customers under a hosting arrangement ▶ The sales proceeds of computer software developed or obtained for internal use that is marketed ▶ New internal-use software developed or obtained that replaces previously existing internal-use software ▶ Computer software that consists of more than one component or module (see Section 3.2.3.1 for discussion on the unit of account for amortization) ▶ Costs incurred to develop a website	▶ Software to be sold, leased, or otherwise marketed as a separate product or as part of a product or process, subject to ASC 985-20 (see Chapter 2) ▶ Software to be used in research and development, subject to ASC 730-10 (see Section 1.6) ▶ Software developed for others under a contractual arrangement, subject to contract accounting standards (see Section 1.7) ▶ Accounting for costs of reengineering activities, which often are associated with new or upgraded software applications subject to ASC 720-45, <i>Business and Technology Reengineering</i>.

Internal-use software has **both** of the following characteristics:

- ▶ The software is acquired, internally developed, or modified solely to meet the entity's internal needs.
- ▶ During the software's development or modification, no substantive plan exists or is being developed to market the software externally (see Section 1.3.1).

BDO INSIGHTS: ACCOUNTING FOR ARTIFICIAL INTELLIGENCE SOFTWARE

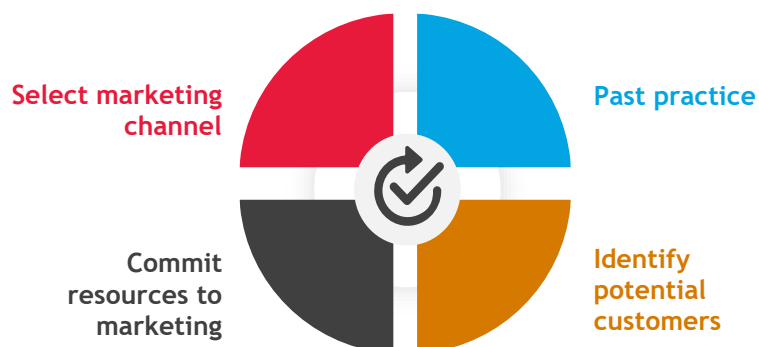
AI solutions, including generative AI, are typically only accessed on a hosted basis and thus meet the definition of internal-use software. Therefore, costs to develop an AI tool will typically be accounted for under ASC 350-40.

An entity may also license an AI solution to use in its own operations. That license is considered internal-use software, and the associated costs are accounted for under ASC 350-40.

1.3.1 Substantive Plan to Market

An entity must first determine whether it has developed a substantive plan to market the software. Only if the answer to that question is no is the project in the scope of guidance on internal-use software.

A substantive plan to market software externally could include the selection of a marketing channel with identified promotional, delivery, billing, and support activities. To be considered a substantive plan, implementation of the plan must be **reasonably possible**. Arrangements providing for the joint development of software for mutual internal use, for example, cost-sharing arrangements and routine market feasibility studies are not substantive plans to market software. Typically, an entity planning to sell or license software externally would identify potential customers and commit resources for marketing activities.



An entity's past practices related to selling software may help determine whether the software is for internal use or subject to a plan to be marketed externally. For example, an entity in the business of selling software often both uses and sells its own software products. The past practice of both using and selling computer software creates a **rebuttable presumption** that any software developed by that entity is intended for sale, lease, or other marketing.

Sometimes an entity that did not plan to license or sell internal-use software to external parties subsequently decides to license or sell that software. Refer to Section 3.3.3 for guidance on accounting considerations for an entity's subsequent decision to market internal-use software to customers.

1.3.2 Scope of ASC 350-40 Versus ASC 985-20



FASB REFERENCES

ASC 350-40-15-5

Costs of computer software that is sold, leased, or otherwise marketed as a separate product or as part of a product or process are within the scope of ASC 985-20. For example, software designed for and embedded in a chip is included in the scope of ASC 985-20 because it is an integral part of the product.

In contrast, software that is used in developing a product, rather than as part of or included in the actual product or service sold, is considered internal-use software. Similarly, if an entity uses software in the production of the product or providing the service but the customer does not acquire the software or the future right to use it, the software is

within the scope of ASC 350-40. For example, for a telecommunications entity selling telephone services, software in the telephone switches is part of its internal equipment that it uses to deliver the services. The software is not acquired or received by the customer and is therefore within the scope of ASC 350-40.

1.4 HOSTING ARRANGEMENT OR CCA



FASB REFERENCES

ASC 350-40-15-4A through 15-4D and ASC 985-20-15-5 through 15-7

ASC 350-40-20: Hosting Arrangement

In connection with accessing and using software products, an arrangement in which the customer of the software does not currently have possession of the software; rather, the customer accesses and uses the software on an as-needed basis.

A hosting arrangement or CCA refers to an arrangement whereby a customer remotely accesses software hosted on the vendor’s or a third-party’s cloud platform servers. In those arrangements, an entity develops or acquires software that is used to provide services – rather than to license the software – to end users or customers.

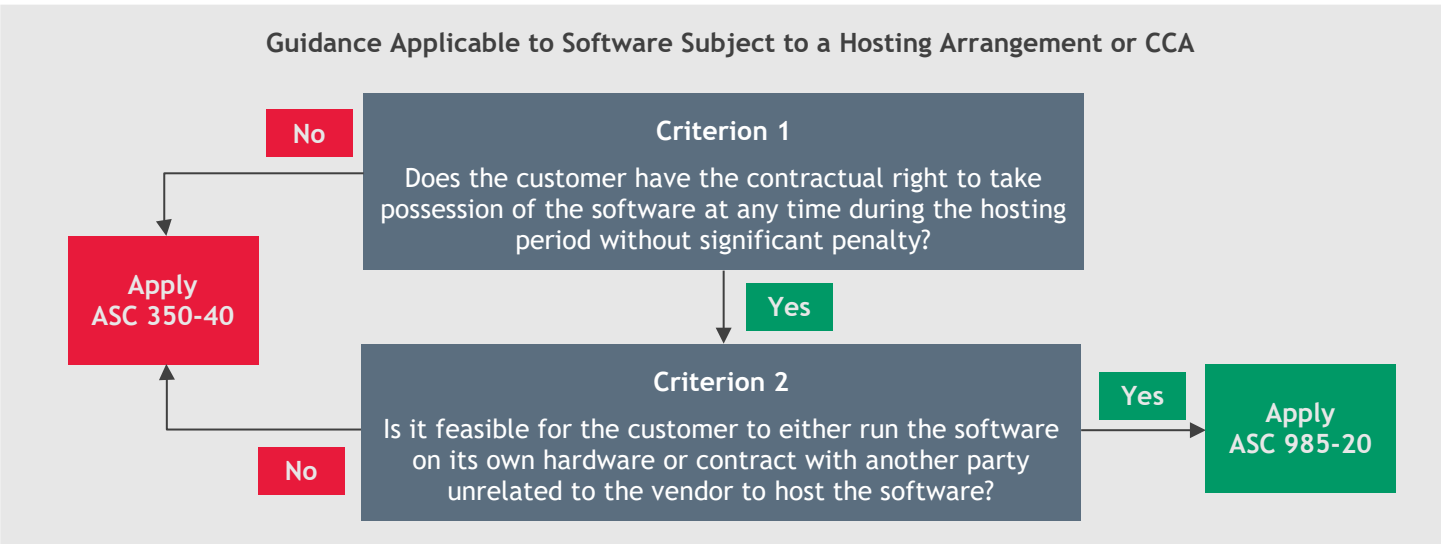
1.4.1 Developer Accounting

Software subject to a hosting arrangement is within the scope of ASC 985-20 if **both** of the following criteria are met:

- ▶ The customer has the contractual right to take possession of the software at **any time** during the hosting period without significant penalty.
- ▶ It is feasible for the customer to either run the software on its own hardware or contract with another party unrelated to the vendor to host the software.

When determining whether the customer incurs a significant penalty to take possession of the software, an entity must consider both the ability to take possession of the software without incurring significant cost, and the ability to use the software separately without a significant diminution in utility or value.

If either of the criteria are not met, then the software is used in providing services and is outside the scope of ASC 985-20. Therefore, the entity applies guidance in ASC 350-40 on internal-use software to account for the related development costs, as illustrated in the following flowchart:



1.4.2 Customer Accounting

The implementation costs of a hosting arrangement or CCA that is a service contract — that is, it does **not** meet both criteria listed in Section 1.4.1 — are accounted for in accordance with ASC 350-40 (see Section 3.5). That guidance follows the same scope and scope exceptions as applicable to internal-use software (see Section 1.3) and specifies:

- ▶ Whether the costs incurred to implement a hosting arrangement or CCA that is a service contract are capitalized
- ▶ Subsequent accounting for a capitalized asset
- ▶ Related presentation requirements

However, the costs associated with a hosting arrangement and the related amortization are presented separately from those associated with internal-use software (see Section 4.2.2.2).

1.5 SOFTWARE ACQUIRED IN A BUSINESS COMBINATION



FASB REFERENCES

ASC 805-10-35-1

An entity may acquire software as part of a business combination. ASC 805, provides guidance on the identification and initial measurement of the identified intangible assets acquired in a business combination (see BDO's Blueprint, [Business Combinations Under ASC 805](#)). After the initial accounting for the business combination, an acquirer subsequently measures and accounts for the assets acquired in accordance with other U.S. GAAP applicable to those assets, depending on their nature. Therefore, an entity recognizes and measures the acquired software at its fair value at the date of acquisition and then applies ASC 985-20 or ASC 350-40 to subsequently account for the software, based on how it intends to use the software.

1.6 SOFTWARE IN RESEARCH AND DEVELOPMENT ACTIVITIES



FASB REFERENCES

ASC 350-40-15-7, ASC 985-20-60-2, ASC 730-10-25-1 through 25-4, and ASC 730-10-55-1(h)

When software for use in research and development activities is purchased or leased, its cost is expensed as incurred unless the software has alternative future use in research and development activities or otherwise.

Additionally, development of software to be used in research and development activities are research and development costs and, therefore, are expensed when incurred.

All costs of developing internal-use computer software (including software developed by third parties like programmer consultants) are expensed as incurred in **either** of the following circumstances:

- ▶ The software is a pilot project (that is, software similar to a pilot plant that is not of a scale economically feasible for commercial production).
- ▶ The software is used in a particular research and development project, regardless of whether the software has alternative future uses.

1.6.1 Research and Development Arrangements



FASB REFERENCES

ASC 985-20-25-12 and ASC 985-20-60-3

For software-development arrangements that are fully or partially funded by a party other than the vendor that is developing the software and for which technological feasibility of the computer software product has **not** been established (see Section 2.2) before entering the arrangement, ASC 730-20, *Research and Development Arrangements*, applies.

In contrast, ASC 730-20 does not apply to funded software-development arrangements in which the technological feasibility of the computer software product has been established before the arrangement is executed because such arrangement is not a research and development project. Instead, if the funding party meets the definition of a customer, the entity applies ASC 606 to those arrangements (see BDO's Blueprint, [Revenue Under ASC 606](#)). See Section 1.7 of this Blueprint for the accounting for the related costs.

If the arrangement is **not** a contract with a customer and capitalization of the software-development costs starts pursuant to ASC 985-20 (see Sections 2.2 and 2.3):

- ▶ Any income from the funding party under a funded software-development arrangement is credited first to (reduces) the development costs previously capitalized.
- ▶ If the income from the funding party exceeds the amount of development costs capitalized, the excess is deferred and credited against future amounts that subsequently qualify for capitalization.
- ▶ Any deferred amount remaining after the project is complete (when the software is available for general release to customers and capitalization has ceased) is credited to income.

BDO INSIGHTS: ACCOUNTING FOR FUNDED SOFTWARE ARRANGEMENTS

As discussed in Section 2.2.1, the technological feasibility of a computer software product is established when the entity has completed all planning, designing, coding and testing activities that are necessary to establish that the product can be produced to meet its design specifications including functions, features, and technical performance requirements, typically evidenced by a working model. Because the time between establishing technological feasibility and completing the software product is typically short, most funded software development arrangements will qualify as research and development projects accounted for under ASC 730-20.

However, if a funded software arrangement is entered after technological feasibility is established but before the software is complete and available for release to customers, the entity must determine whether the arrangement is in the scope of ASC 606 rather than ASC 985-20, as the accounting treatment differs. Specifically, the entity must determine whether the arrangement represents the development and sale of intellectual property to a customer. Factors to consider when making that determination may include, but are not limited to:

- ▶ Whether the entity will retain the right to license or sell the software
- ▶ Whether modifications to the software's features and functionality were made at the funding party's direction
- ▶ Whether the entity has a past practice of entering similar arrangements

Determining whether a funded software arrangement meets the definition of a contract with a customer requires the application of professional judgment based on the facts and circumstances.

1.7 SOFTWARE COSTS IN CONTRACTS WITH CUSTOMERS



FASB REFERENCES

ASC 340-40-15-3, ASC 340-40-25-5 through 25-8, ASC 340-40-35-1, and ASC 340-40-55-5 through 55-9

An entity may be engaged to develop software that it will deliver to a customer in a revenue arrangement accounted for under ASC 606. In those situations, the developer applies ASC 340-40 to software costs unless those costs fall within the scope of ASC 350-40 or ASC 985-20 (see Sections 1.2 and 1.3). For software costs incurred in fulfilling a contract with a customer that are within the scope of ASC 340-40, an entity recognizes an asset if those costs meet **all** the following criteria:

- ▶ The costs relate directly to a contract or anticipated contract the entity can specifically identify. For example, costs relating to services provided under renewal of an existing contract or costs of designing an asset transferred under a specific contract that has not yet been approved but is expected to be approved would be considered costs to fulfill a contract.
- ▶ The costs generate or enhance resources of the entity that will be used in satisfying performance obligations in the future.
- ▶ The costs are expected to be recovered.

An asset, if recognized, is amortized on a systematic basis that is consistent with the transfer to the customer of the goods or services to which the asset relates (see Section 7.7 of BDO's Blueprint, [Revenue Recognition Under ASC 606](#)).

However, if an entity develops software that it will use to deliver services to a customer in a revenue arrangement, typically that software will be considered internal-use software, and the related costs will be accounted for under ASC 350-40 as illustrated in the following example:

EXAMPLE 1-1: [ADAPTED FROM ASC 340-40-55-5 THROUGH 55-9] SOFTWARE COSTS IN CONTRACTS WITH CUSTOMERS

FACTS

An entity enters a contract with a customer to manage the customer's information technology data center for five years. The contract is renewable for subsequent one-year periods. The average customer term is seven years. The entity incurs the following costs:

- ▶ An entity enters a contract with a customer to manage the customer's information technology data center for five years. The contract is renewable for subsequent one-year periods. The average customer term is seven years.
- ▶ Before providing the services, the entity designed and built a technology platform for its internal use that interfaces with the customer's systems. That platform is not transferred to the customer but will be used to deliver services to the customer.
- ▶ The entity incurred the following initial costs to set up the technology platform:

Design services	\$ 20,000
Hardware	60,000
Software	45,000
Data migration and testing	50,000
Total costs incurred	\$ 175,000

CONCLUSION

The entity accounts for the initial setup costs as follows:

- ▶ Hardware costs: accounted for in accordance with ASC 360, *Property, Plant, and Equipment*
- ▶ Software costs: accounted for in accordance with ASC 350-40
- ▶ Design services, data migration, and testing: assessed in accordance with ASC 340-40 to determine whether an asset can be recognized for the costs to fulfill the contract. Any resulting asset would be amortized on a systematic basis over a seven-year period.

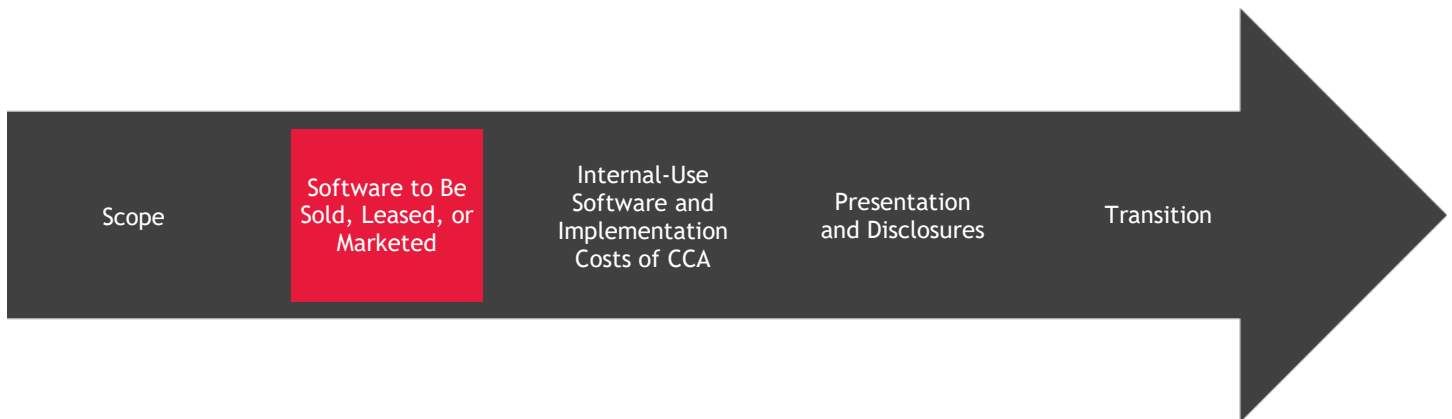
ANALYSIS

The initial set-up costs relate primarily to activities to fulfill the contract but do not transfer goods or services to the customer. The entity accounts for the initial setup costs as follows:

- ▶ Hardware costs: accounted for in accordance with ASC 360
- ▶ Software costs: accounted for in accordance with ASC 350-40
- ▶ Design services, data migration, and testing: assessed in accordance with ASC 340-40 to determine whether an asset can be recognized for the costs to fulfill the contract. Any resulting asset would be amortized on a systematic basis over the seven-year period (that is, the five-year contract term and two anticipated one-year renewal periods) the entity expects to provide services related to the data center.



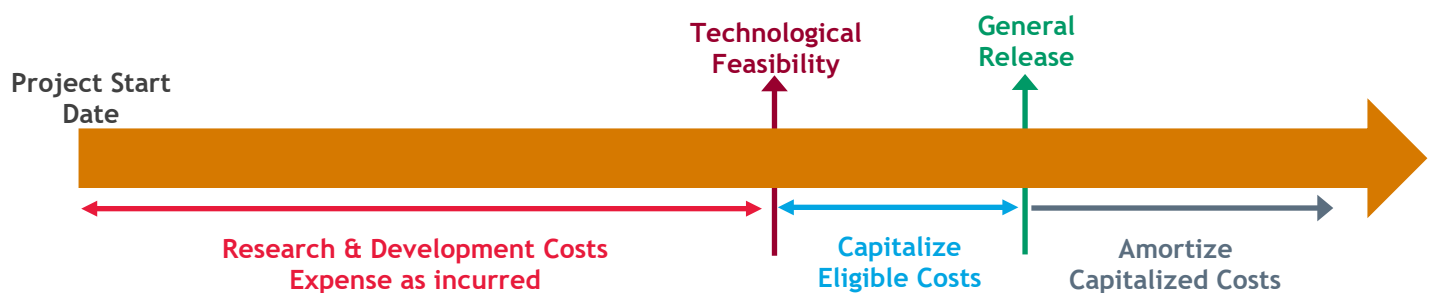
Chapter 2: Software to Be Sold, Leased, or Marketed



2.1 OVERVIEW

ASC 985-20 provides guidance for accounting and reporting for the costs of computer software to be sold, leased, or otherwise marketed as a separate product or as part of a product or process (externally marketed software), whether internally developed and produced or purchased. ASC 985-20 specifies whether an entity must expense or capitalize the software development costs based on whether the costs are incurred before or after the product reaches technological feasibility:

- ▶ Costs incurred before reaching technological feasibility are considered research and development costs and expensed when incurred.
- ▶ Costs to develop, produce or purchase externally marketed software are capitalized after technological feasibility is reached. Capitalization of eligible costs ceases once the product is available for general release.



Additionally, ASC 985-20 provides guidance on subsequent accounting for capitalized costs and accounting for costs incurred after the product is available for general release and product enhancements.

2.2 ESTABLISHING TECHNOLOGICAL FEASIBILITY



FASB REFERENCES

ASC 985-20-25-1 through 25-12 and ASC 985-20-55-4 through 55-14

ASC 985-20-20: Coding

Generating detailed instructions in a computer language to carry out the requirements described in the detail program design. The coding of a computer software product may begin before, concurrent with, or after the completion of the detail program design.

ASC 985-20-20: Product Design

A logical representation of all product functions in sufficient detail to serve as product specifications.

ASC 985-20-20: Testing

Performing the steps necessary to determine whether the coded computer software product meets function, feature, and technical performance requirements set forth in the product design.

All costs incurred to establish the technological feasibility of a computer software product to be sold, leased, or otherwise marketed are research and development costs and expensed when incurred under ASC 730.

The technological feasibility of a computer software product is established when the entity has completed all planning, designing, coding, and testing activities necessary to establish that the product can be produced to meet its design specifications including functions, features and technical performance requirements.

To obtain evidence that technological feasibility has been established, an entity must have performed specific minimum activities by completing **either** a detail program design or a working model:

- ▶ **Detail Program Design Approach:** This approach is typically applied if the process of creating the computer software product includes a detail program design.
- ▶ **Working Model Approach:** This approach is typically applied if the process of creating the computer software product does not include a detail program design.

BDO INSIGHTS: TWO APPROACHES TO EVIDENCE TECHNOLOGICAL FEASIBILITY

Determining which approach provides evidence that technological feasibility has been established is **not** an accounting policy election. Rather, an entity determines which approach provides better evidence that technological feasibility was established based on its software development process for a project.

In practice, many software entities that use agile or iterative software development methods define technological feasibility as the creation of a working model, which often occurs late in the software development cycle.

The two approaches based on a detail program design or working model and related activities are described in the following diagram:

DETAILED PROGRAM DESIGN APPROACH

A detailed program design is defined as the “*detail design of a computer software product that takes product function, feature, and technical requirements to their most detailed, logical form and is ready for coding.*”

The process of creating the computer software product **includes** a **detail program design** and all **three** specified **criteria** are met.

Criterion 1

The product design and the detail program design have been completed, **and** the entity has established that the necessary skills, hardware, and software technology are available to the entity to produce the product.



Criterion 2

The completeness of the detail program design and its consistency with the product design have been confirmed by documenting and tracing the detail program design to product specifications.



Criterion 3

The detail program design has been reviewed for high-risk development issues (for example, novel, unique, unproven functions and features or technological innovations) **and** any uncertainties related to identified high-risk development issues have been resolved through coding and testing.

WORKING MODEL APPROACH

A working model is defined as “*an operative version of the computer software product that is completed in the same software language as the product to be ultimately marketed, performs all the major functions planned for the product, and is ready for initial customer testing (usually identified as beta testing).*”

The process of creating the computer software product **does not** include a detail program design and **both two** specified **criteria** are met.

Criterion 1

A product design and a working model of the software product have been completed.



Criterion 2

The completeness of the working model and its consistency with the product design have been confirmed by testing.

An entity begins capitalizing eligible costs (see Section 2.3) once technological feasibility is established under either of the two approaches.

One purpose of ASC 985-20 is to identify an objective point in the software product process at which research and development activities end, and production activities begin. To comply with that objective, management **cannot** require more stringent criteria or modify the specified criteria or impose additional criteria to begin capitalizing software production costs.

BDO INSIGHTS: DETERMINING WHETHER TECHNOLOGICAL FEASIBILITY HAS BEEN ESTABLISHED

Management may have its own internal policies (such as definitions or thresholds) to determine whether to advance a software project with additional funding. While those internal policies may help management make business decisions, an entity must apply ASC 985-20 to determine whether technological feasibility for a product has been established. Any other non-GAAP internal threshold or policy **cannot** be used as a substitute for technological feasibility and determine whether software development costs are expensed or capitalized.

2.2.1 Interaction Between the Detail Program Design Approach and Working Model Approach

An entity whose software product process fits the description in the detail program design approach applies the three technological feasibility criteria under that approach. However, if all three criteria in the detail program design approach are not met until a working model is complete, then an entity begins capitalizing the eligible costs only upon completing the working model and satisfying the two criteria in the working model approach.

2.2.2 Working Model

 FASB REFERENCES

ASC 985-20-55-8 and 55-9
ASC 985-20-20: Working Model
An operative version of the computer software product that is completed in the same software language as the product to be ultimately marketed, performs all the major functions planned for the product, and is ready for initial customer testing (usually identified as beta testing).

Some entities in the software industry use the term “working model” to mean a prototype in which critical parts of the product have been coded or written in pseudocode (that is, not computing language). However, such an industry definition of working model does not meet the criteria specified in the guidance under the Working Model Approach. To meet the criteria specified in the guidance, a working model must meet **all** the following conditions:

Working Model Must Be:	Operative
	In the same language as the product that will be marketed
	Complete with all the major functions that were planned for the product
	Ready for initial customer testing

BDO INSIGHTS: WORKING MODEL — BETA VERSION

A beta version of a software product is a nearly complete product in the final phase of product development that is released to a limited group of real users or customers to identify bugs, gather user feedback, and assess overall usability before launching the product in the wider market.

While the guidance states that a working model is **usually** a beta version of a software product, a beta version is **not** always the working model. A prior version of a software product may meet the definition of a working model and the conditions discussed above and, hence, be a complete working model that evidences the establishment of technological feasibility.

Additionally, an entity that does not undertake beta testing, for example, when using an agile software development method, still has a working model when it meets the conditions above.

In other words, while a beta version may indicate that technological feasibility has been established, it is **not a bright line** in determining whether technological feasibility has been established.

2.2.2.1 Linear and Nonlinear (Agile) Software Development Methods

A linear stage-based development method involves a detail program design, a development plan, and a sequential approach to coding. Historically, software was developed using linear methods such as the waterfall method. These methods are suitable in the development of some software but are not considered to be adaptive to changes in software requirements. Because each phase under a linear method must be completed in sequential order before moving to the next phase, it may be challenging and costly to go back to a previous phase and make changes. Specifically, it may be difficult to outline all the performance requirements at the beginning of the software project, and significant changes in later phases can be challenging and costly to incorporate.

To overcome the challenges with linear methods, other software development methods, such as the agile method, have become more prevalent. An agile software development method breaks a software project down into smaller efforts (or sprints) to develop a specific function or feature within the project with continuous incremental and iterative adaptations throughout the development process. The agile method results in individual development and continuous changes to product features and functionalities. The iterative environment makes the software project more manageable and allows an entity to quickly incorporate and develop changes.

BDO INSIGHTS: AGILE SOFTWARE DEVELOPMENT METHOD — WORKING MODEL

Most entities now use agile or iterative software development methods rather than linear stage-based software development methods. In practice, entities that use agile software development methods generally do not complete a detailed program design sufficient to establish technological feasibility and therefore use the working model approach to establish technological feasibility. Because a working model is usually complete much later in the development process, the development costs eligible for capitalization are often minimal, and most development costs are expensed as research and development costs.

2.2.3 Technological Feasibility of the Product as a Whole



FASB REFERENCES

ASC 985-20-55-7

When a product comprises various modules that are not separately sellable, technological feasibility is established for the product as a whole and not on a module-by-module basis. The detail program design or the working model of the entire product (all modules linked together) must be completed before capitalization begins.

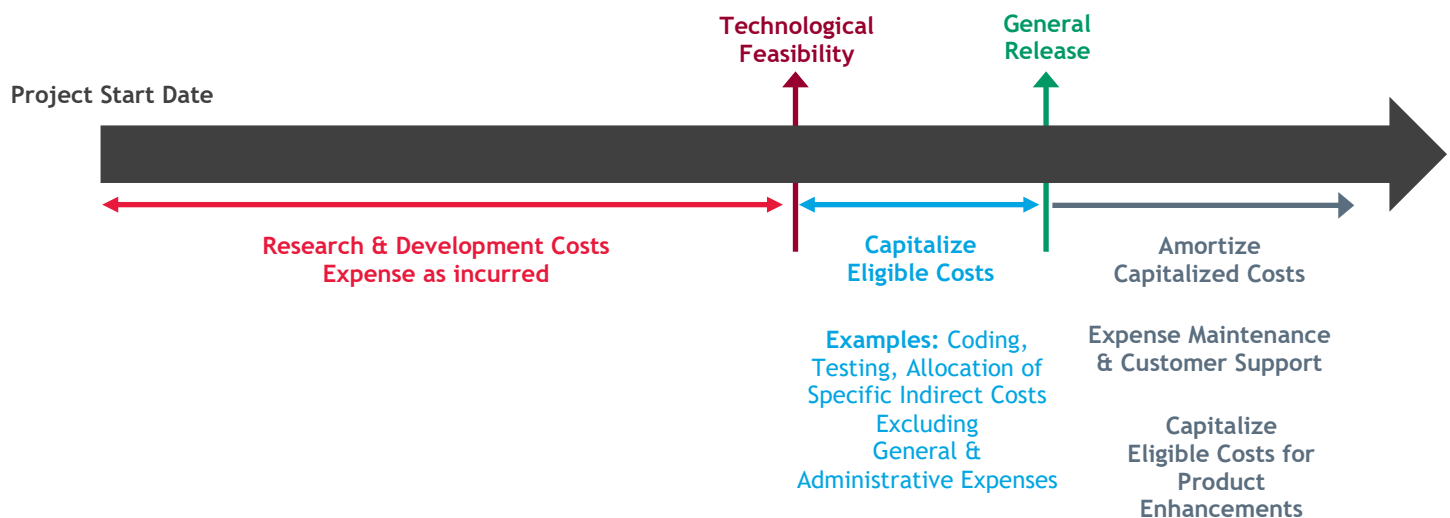
2.3 CAPITALIZABLE COSTS



FASB REFERENCES

ASC 985-20-25-3 through 25-6 and ASC 985-20-55-4 through 55-6

This section provides guidance on whether an entity must expense or capitalize different types of software development costs. The following diagram provides an overview of the guidance discussed in this section:



BDO INSIGHTS: SOFTWARE DEVELOPMENT COSTS – EXPENSE VERSUS CAPITALIZE

ASC 985-20 does not allow a capitalization threshold specifying the minimum amount of costs that must be incurred before capitalizing those costs. However, in practice many entities establish such a threshold for administrative convenience when expensing such amounts would not be material to the financial statements as a whole. If an entity elects to establish a capitalization threshold related to software enhancements, it must apply judgment to determine an appropriate capitalization threshold. In addition, the entity must establish processes and controls to evaluate the impact of such policy on the financial statements.

2.3.1 Production Costs of Computer Software



FASB REFERENCES

ASC 985-20-25-3, ASC 985-20-25-5 through 25-6, and ASC 985-20-55-4 through 55-6

ASC 985-330-20: Product Masters

A completed version, ready for copying, of the computer software product, the documentation, and the training materials that are to be sold, leased, or otherwise marketed.

An entity capitalizes the costs of producing product masters that are incurred after establishing technological feasibility. Those costs include coding and testing performed after establishing technological feasibility.

Additionally, an entity may capitalize an allocated amount of indirect costs, for example, overhead related to programmers and the facilities they occupy. However, an allocation of general and administrative expenses is **not** appropriate because those costs relate to the period in which they are incurred.

2.3.1.1 Costs Incurred After General Release to Customers

An entity ceases capitalizing computer software costs when the product is available for general release to customers. Costs of maintenance and customer support (see Section 2.3.1.3) are expensed when the related revenue is recognized or when those costs are incurred, whichever occurs first.

While the term “general release” is not defined, it is generally understood as the point when the software product is available for purchase by customers.

2.3.1.2 Computer Software That is an Integral Part of a Product or Process



FASB REFERENCES

ASC 985-20-25-4

An entity cannot capitalize software production costs for computer software that is to be used as an integral part of a product or process until **both** of the following conditions have been met:

- ▶ Technological feasibility has been established for the software.
- ▶ All research and development activities for the other components of the product or process have been completed.

2.3.1.3 Maintenance and Customer Support



FASB REFERENCES

ASC 985-20-25-6, ASC 985-20-55-11 through 55-12, and ASC 985-20-20: Maintenance, Customer Support and Product Enhancements

The guidance includes the following definitions of maintenance, customer support, and product enhancements:

 Maintenance	<i>“Activities undertaken after the product is available for general release to customers to correct errors or keep the product updated with current information. Those activities include routine changes and additions.”</i>
 Customer Support	<i>“Services performed by an entity to assist customers in their use of software products. Those services include any installation assistance, training classes, telephone question and answer services, newsletters, on-site visits, and software or data modifications.”</i>
 Product Enhancements	<i>“Improvements to an existing product that are intended to extend the life or improve significantly the marketability of the original product. Enhancements normally require a product design and may require a redesign of all or part of the existing product.” (See Section 2.5).</i>

Determining whether an activity is maintenance, customer support, or a product enhancement requires judgment. An entity must evaluate its circumstances and intentions, as well as the definitions for each activity. See Section 2.5 for discussion on product enhancements.

An entity may promise to keep software current with revisions in other related software or the hardware on which it runs and incur costs in connection with that service. For example, developers of mobile applications may commit to revising the software as the operating systems running on mobile phones are updated so that the software continues to function as designed. That activity often meets the definition of maintenance because it updates the product with current information rather than providing additional functionality.

The costs of maintenance and customer support are expensed when related revenue is recognized or when those costs are incurred, whichever occurs first.

2.3.1.4 Inventory Costs for Duplicating the Product Masters

 FASB REFERENCES

ASC 985-20-25-11, ASC 985-330-25-1, and ASC 985-330-40-1

The costs incurred for duplicating computer software, documentation, and training materials from the product masters, and for physically packaging the product for distribution are:

- ▶ Capitalized as inventory on a unit-specific basis.
- ▶ Expensed in cost of sales when revenue from the sale of those units is recognized.

2.3.2 Purchased Computer Software

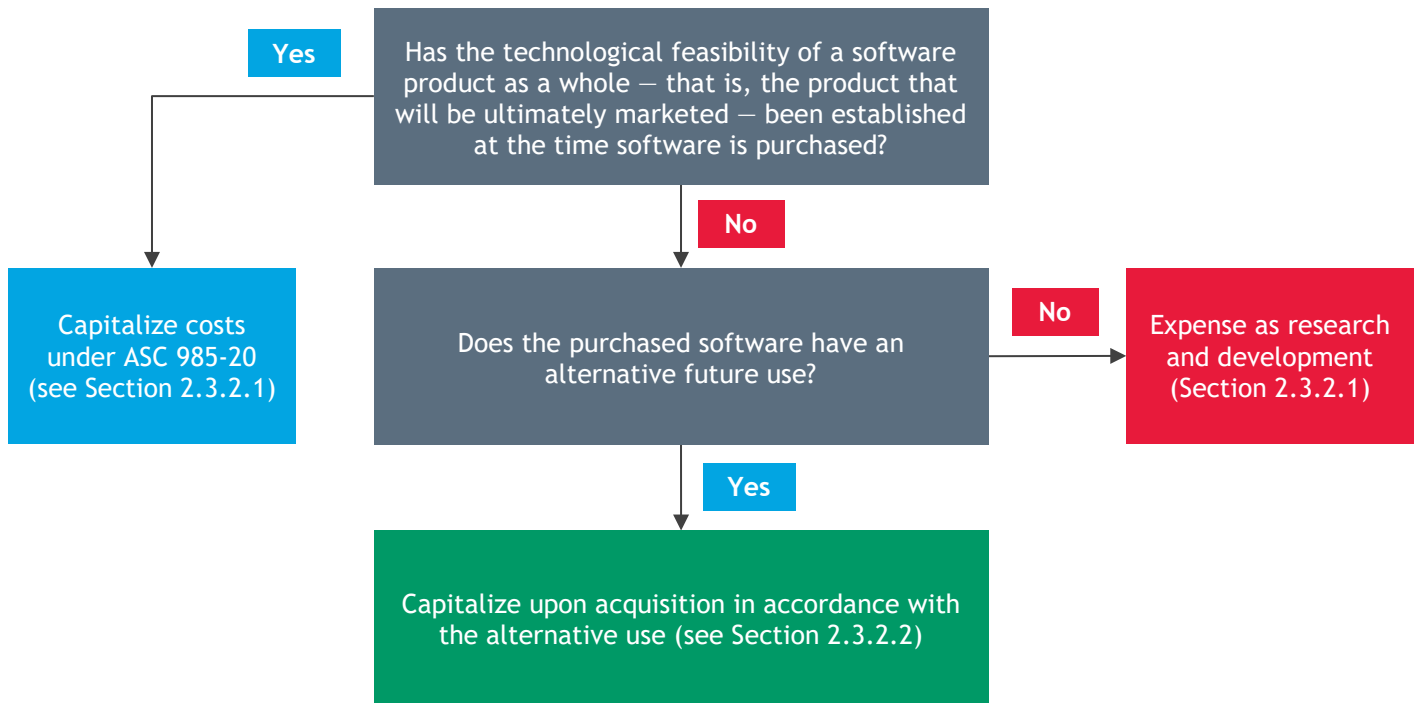
 FASB REFERENCES

ASC 985-20-25-7 through 25-10, ASC 985-20-55-13 through 55-14

An entity may purchase computer software to be resold. Further, an entity may modify or integrate the purchased software with another product or process. Whether the costs of purchased computer software are expensed or capitalized depends on whether:

- ▶ The technological feasibility of a software product as a whole – that is, the product that will be marketed – has been established at the time software is purchased **or**
- ▶ The purchased software has an alternative future use.

The following diagram illustrates the guidance for costs of purchased computer software:



2.3.2.1 Purchased Software with No Alternative Future Use

An entity accounts for the cost of purchased computer software to be sold, leased, or otherwise marketed that has **no alternative future use** in the same manner as the costs incurred to develop such software internally (see Section 2.3.1):

- ▶ An entity capitalizes the total cost of purchased software that has no alternative future use if **both** of the following criteria are met at the time of purchase:
 - Technological feasibility has been established for the software.
 - All research and development activities for the other components of the product or process have been completed.
- ▶ If either of the above criteria is **not** met at the time of purchase, the cost of purchased software is expensed as research and development.

For example, if the technological feasibility of a software product as a whole – that is, the product that will be marketed – has been established at the time software is purchased, the cost of the purchased software is capitalized and further accounted for in accordance with the other provisions of ASC 985-20.

2.3.2.1.1 Purchased Software to be Integrated into Another Product



FASB REFERENCES

ASC 985-20-25-9 and ASC 985-20-55-13

An entity may purchase software that will be integrated into another software or hardware product. The cost of software purchased to be integrated with another product or process is capitalized when **both**:

- ▶ Technological feasibility is established for the software component.
- ▶ All research and development activities for the other components of the product or process are completed at the time of purchase.

If the purchased computer software has no alternative future use, its costs can be capitalized only if the technological feasibility of the product to be marketed has been established at the time of purchase. Factors such as the timing of receipt or the status of hardware and internal software development may be important in determining whether technological feasibility is established at the time of purchase.

2.3.2.2 Purchased Software with Alternative Future Use



FASB REFERENCES

ASC 985-20-25-10

An entity may purchase software that it intends either to resell or to integrate into a product or other software and that software (or the final product) is not yet technologically feasible. However, if that software has an alternative future use, the entity must capitalize the cost of the purchased software when it acquires the software and account for that software in accordance with its use.

EXAMPLE 2-1: [ADAPTED FROM ASC 985-20-55-14] SOFTWARE PURCHASED BEFORE TECHNOLOGICAL FEASIBILITY ESTABLISHED**FACTS**

An entity purchases software for \$100,000 before technological feasibility of the software has been established. The entity plans to incorporate the purchased software into a software product it is developing but could also resell the purchased software for \$75,000.

CONCLUSION

The entity accounts for the cost of the software as follows:

- ▶ \$25,000 is expensed to research and development.
- ▶ \$75,000 is capitalized.

ANALYSIS

The entity determines that the software has an alternative future use at the date of purchase because it can resell the software for \$75,000. Therefore, the entity capitalizes \$75,000. Furthermore, based on the alternative use of the software and whether technological feasibility is established, the entity would classify the \$75,000 as follows:

- ▶ If the software product reaches technological feasibility, \$75,000 is included in the cost of the software product.
- ▶ If the technological feasibility of the software product is never established, the \$75,000 is classified as inventory until the purchased software is resold.

2.4 SUBSEQUENT ACCOUNTING FOR CAPITALIZED SOFTWARE COSTS



FASB REFERENCES

ASC 985-20-35-1 through 35-4 and ASC 985-20-55-15 through 55-17

Capitalized software costs are amortized over the estimated remaining economic life of the software, beginning once the software project is available for general release to customers (see Section 2.4.1). In addition, at each balance sheet date, an entity must evaluate the remaining unamortized costs for impairment by applying a net realizable value test to compare the unamortized capitalized costs of a computer software product to the net realizable value of that product (see Section 2.4.2).

2.4.1 Amortization of Capitalized Software Costs



FASB REFERENCES

ASC 985-20-35-1 through 35-4 and ASC 985-20-55-15 through 55-17

An entity amortizes capitalized software costs on a product-by-product basis. Amortization of capitalized software costs begins when the product is available for general release to customers. As discussed in Section 2.3.1.1, the term “general release” is generally understood as the point when the software product is available for purchase by customers.

The annual amortization is the **greater** of the amounts computed using the following:

- ▶ The ratio that current gross revenues for a product bear to the total of current and anticipated future gross revenues for that product.
- ▶ The straight-line method over the remaining estimated economic life of the product, including the current reporting period – that is, the unamortized cost of the product is divided by its remaining life, including the current year.

The FASB requires this “greater of” calculation because the net realizable value test (which considers future revenues and costs) also must be applied to capitalized software costs. Therefore, in the first part of the calculation, amortization is based on estimated future revenues. However, because of the uncertainties involved in estimating revenue, amortization **cannot** be less than straight-line amortization over the product’s remaining estimated economic life.

Amortization for any asset is based on estimates of future events, and software is no exception. In determining the annual amortization, estimates of future revenues or the remaining economic life for a product may change over the period in which the software product is being amortized. An entity must use the most recent information to determine whether to change an estimate.

2.4.2 Impairment of Capitalized Software Costs: Net Realizable Value Test



FASB REFERENCES

ASC 985-20-35-4

At each balance sheet date, an entity must apply a net realizable value test to capitalized software costs to compare the unamortized capitalized costs of a computer software product to the net realizable value of that product. The net realizable value is the estimated future gross revenues from that product reduced by the estimated future costs of completing and disposing of that product, including the costs of performing maintenance and customer support required to satisfy the entity's responsibilities set forth at the time of sale.

The amount by which the unamortized capitalized costs of a computer software product exceed the net realizable value of that asset is written off.

$$\text{Impairment Write Off} = \text{Unamortized Capitalized Costs} - \text{Net Realizable Value}$$



The resulting amount of capitalized computer software costs (after the write down to net realizable value) at the close of an annual fiscal period is considered the cost for subsequent accounting purposes. The new basis of the capitalized computer software costs is **not** subsequently restored if the value increases in future periods.

BDO INSIGHTS: NET REALIZABLE VALUE DISCOUNTED OR UNDISCOUNTED

The guidance does not specify whether the net realizable value of a software product is determined based on discounted or undiscounted estimates of future revenues and costs. In the absence of specific guidance requiring the calculation on a discounted basis, we believe that an entity determines the net realizable value of a software product on an undiscounted basis. This approach is consistent with how net realizable value is determined under ASC 330, *Inventory*.

BDO INSIGHTS: NET REALIZABLE VALUE — ESTIMATES OF FUTURE REVENUES AND COSTS

Determining the net realizable value of a software product requires judgment based on the facts and circumstances. The guidance states that the net realizable value of a software product is estimated based on:

- ▶ Future gross revenues from that product
- ▶ Future costs of completing and disposing of that product, including maintenance and customer support.

Arrangements to provide post-contract support (PCS) typically cover future upgrades to, maintenance of, and customer support for a software product. We believe it is appropriate to include expected revenues from and costs of any PCS arrangements in determining the net realizable value of a software product.

Additionally, if an entity uses the same software product in hosting arrangements with customers, we believe it is appropriate to include revenues from the hosting arrangements and related costs in determining the net realizable value of that software product.

2.5 PRODUCT ENHANCEMENTS**FASB REFERENCES**

ASC 985-20-55-18 through 55-22 and ASC 985-20-20: Product Enhancements

Product enhancements are defined as *“improvements to an existing product that are intended **to extend the life or improve significantly the marketability** of the original product. Enhancements normally require a product design and may require a redesign of all or part of the existing product.”* (Emphasis added).

Costs of product enhancements are usually not incurred until after the related software product has been made available for sale to the public.

Product enhancements are specifically included in the scope of ASC 985-20 and are subject to the same requirements as any other software product. Costs incurred for product enhancements are expensed as research and development until the technological feasibility (see Sections 2.2 and 2.5.1) of the enhancement has been established.

If the original product will no longer be marketed, any unamortized cost of the original product is included with the cost of the enhancement for purposes of applying the net realizable value test and amortization (see Section 2.4). If the original product will remain on the market along with the enhancement, the unamortized cost of the original product is allocated between the original product and the enhancement.

All costs of a product enhancement, including any costs carried over or allocated from the original product, are amortized over the enhancement's estimated useful life (see Section 2.5.2).

BDO INSIGHTS: PRODUCT ENHANCEMENTS VERSUS MAINTENANCE

Once a software product has been released for sale, an entity often incurs costs related to maintenance, customer support, and product enhancements. Costs related to maintenance and customer support are expensed when incurred while costs to develop product enhancements may be eligible for capitalization.

Costs that maintain the promised functionality of an existing product are maintenance costs, whereas costs that extend the life or significantly improve the marketability of an existing software product are product enhancement costs that qualify for capitalization if technological feasibility is met (see Section 2.5.1). While the guidance does not define the term “marketability,” we believe it refers to a product's ability to generate a new revenue stream or expand the market for the product or its appeal to potential customers.

For example, an entity that develops and sells video game licenses often incurs costs after general release of the game to maintain compatibility between its software and a particular game console hardware. Those costs are typically considered maintenance costs and expensed. Conversely, if the entity modifies its existing software to run on a different type of game console hardware resulting in a new revenue stream from the software product, those costs are considered product enhancement and considered for capitalization (see Section 2.5.1).

It may be challenging to discern whether a change to the software qualifies as an enhancement. Determining whether the costs associated with subsequent changes to the software should be expensed as maintenance or capitalized as an enhancement requires the application of professional judgment based on the facts and circumstances.

2.5.1 Technological Feasibility of a Product Enhancement



FASB REFERENCES

ASC 985-20-55-20 through 55-22

The technological feasibility criteria (see Section 2.2) that must be met for the original product must also be met for a product enhancement to capitalize the related costs of development.

Notably, technological feasibility may be more easily established for a product enhancement than for a new product, and, therefore, capitalization of costs may begin relatively earlier in the software development process. For example, an enhancement that adds one new function to an already successful product may require only minor modifications to the original product's detail program design to establish technological feasibility. Similarly, software that is ported (that is, made available for different hardware) may not require a new detail program design. Capitalization of the enhancement costs may begin once any high-risk development issues have been resolved.

2.5.2 Useful Life of a Product Enhancement



FASB REFERENCES

ASC 985-20-55-19

A product enhancement is treated as a separate unit of account with its own estimated useful life. The estimated useful life of a product enhancement is the estimated life of the enhancement.

The following are **not** the estimated useful life of a product enhancement:

- ▶ The remaining life of the original product
- ▶ The remaining life of the original product for any costs of the original product included in the enhancement and the estimated life of the enhancement for all other costs.

2.6 ISSUES ARISING AFTER ESTABLISHING TECHNOLOGICAL FEASIBILITY



FASB REFERENCES

ASC 985-20-25-2, ASC 985-20-55-10, and ASC 250-10-45-17

Technological feasibility is not established until all high-risk development issues are resolved through coding and testing. However, a high-risk development issue may arise after an entity had previously established technological feasibility (see Section 2.2) for the software, but prior to general release.

Such a situation is accounted for as a change in estimate in accordance with ASC 250, *Accounting Changes and Error Corrections* (see BDO's publication, [Accounting Changes and Error Corrections](#)). That guidance states that changes in estimates result from new information. The discovery of a high-risk development issue after an entity originally concluded technological feasibility was established meets that definition.

Accordingly, if an entity identifies high-risk development issues after it had previously established technological feasibility for a software product, it:

- ▶ Expenses any previously capitalized costs for that product as research and development costs in the period in which the development issues are identified.
- ▶ Expenses any additional development costs as incurred as research and development costs until the criteria for technological feasibility (see Section 2.2) are met, which includes resolving the newly identified high-risk development issues.

2.7 MODIFICATIONS TO FEATURES ORIGINALLY PLANNED FOR THE PRODUCT



FASB REFERENCES

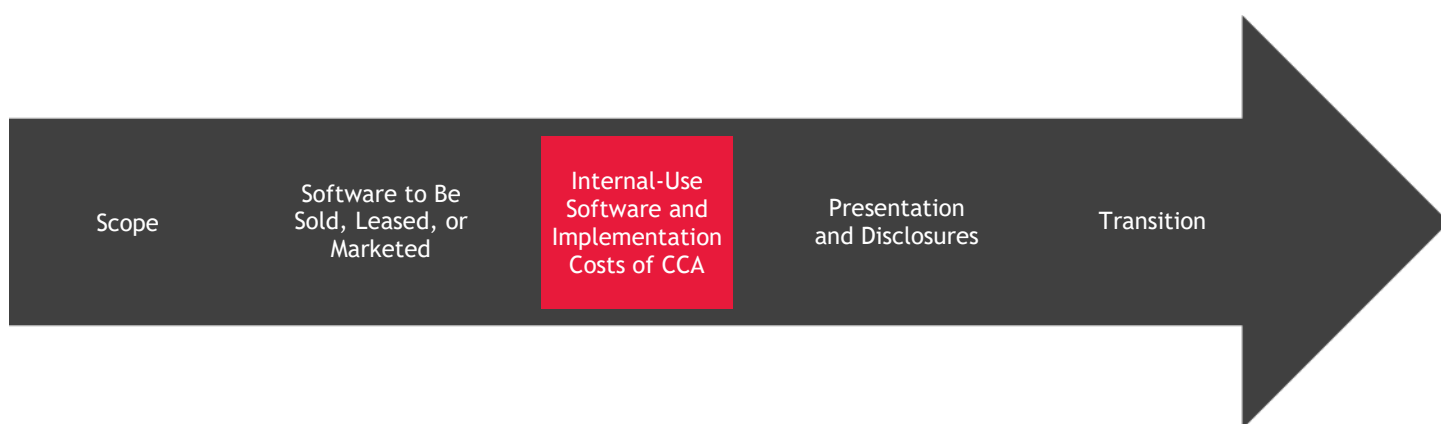
ASC 985-20-55-17

An entity may subsequently decide to omit features or functions that were originally included in the software product design. If the completed software product does not include all features that had originally been planned, the accounting depends on how critical those omitted features or functions are to the functionality of the software.

- ▶ If the product is sellable without the features that were dropped, no specific accounting is required. Costs are capitalized once technological feasibility is met (see Section 2.3).
- ▶ If the product is not sellable without the dropped features, the technological feasibility of the product is not established (see Section 2.2).



Chapter 3: Internal-Use Software and Implementation Costs of CCA



3.1 OVERVIEW

ASC 350-40 provides guidance for accounting and reporting for the costs of internal-use software, including examples that illustrate whether computer software is in scope or not. ASC 350-40 specifies whether an entity must expense or capitalize the software development costs based on whether the costs are incurred before or after a specified recognition threshold. That guidance applies to website development costs as well. Additionally, ASC 350-40 provides guidance on subsequent accounting for capitalized costs and accounting for software that previously was intended for internal use but subsequently was marketed for sale.

Additionally, ASC 350-40 applies to costs to implement a hosting arrangement or CCA that is a service contract (for example, SaaS).



FASB UPDATE: AMENDMENTS TO THE INTERNAL-USE SOFTWARE GUIDANCE IN ASC 350-40

The FASB issued ASU 2025-06 in September 2025 to amend the internal-use software guidance in ASC 350-40. The amendments modernize the internal-use software guidance to accommodate the evolution in software development methods from a linear development method (for example, waterfall) to a nonlinear development method (for example, agile). Before the issuance of ASU 2025-06, ASC 350-40 required an entity to consider project stages in determining whether a software development cost for internal-use software is capitalized or expensed. That model aligned with linear, stage-based software development methods like the waterfall approach but was difficult to apply to agile or iterative software development methods which are more prevalent in current practice. See Section 2.2.2.1 for a discussion of linear and nonlinear (agile) software development methods.

ASU 2025-06 amends ASC 350-40 by removing all references to project stages and providing a principles-based recognition threshold to determine capitalizable software costs. Eliminating outdated references to project stages is expected to improve operability of the guidance across different software development methods.

Additionally, ASU 2025-06 supersedes the previous guidance for website development costs in ASC 350-50 and incorporates the recognition requirements for website-specific development costs into ASC 350-40, aligning it with the new guidance for internal-use software.

This publication assumes that an entity has adopted ASU 2025-06. See Chapter 5 for transition guidance.

3.2 CAPITALIZATION CRITERIA AND CAPITALIZABLE COSTS



FASB REFERENCES

ASC 350-40-05-1D through 05-1E and ASC 350-40-25-1

An entity is required to capitalize specific costs incurred for computer software developed or obtained for internal use if the capitalization criteria in Section 3.2.1 are met. An entity must expense all internal and external costs incurred before meeting the capitalization criteria as they are incurred. Section 3.2.2 specifies the types of costs that are capitalized after the capitalization criteria are met.

3.2.1 Capitalization Criteria



FASB REFERENCES

ASC 350-40-25-12 through 25-14

ASC 350-40-20: Probable

The future event or events are likely to occur.

An entity begins capitalizing internal-use software costs when **both** of the following occur:



Management, with the relevant authority, implicitly or explicitly **authorizes and commits** funding for the software project.

Examples of authorization and commitment to funding a software project include the execution of a contract with a third party to develop the software, approval of expenditures related to internal development or a commitment to obtain the software from a third party.



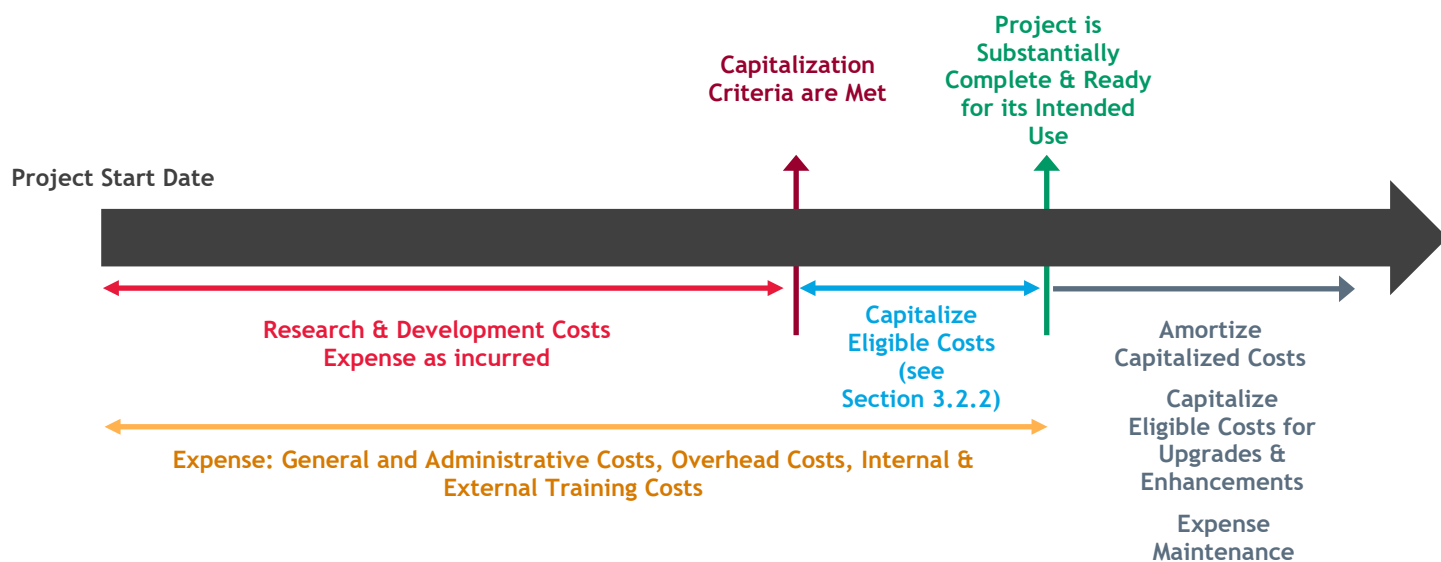
It is **probable** the project will be completed, and the software will be used to perform the function intended (referred to as “**probable-to-complete recognition threshold**”).

In evaluating whether the probable-to-complete recognition threshold has been met, an entity assesses whether there is significant uncertainty associated with the development activities of the software, referred to as “**significant development uncertainty**.” (see Section 3.2.1.1).

If the capitalization criteria are no longer met for software being developed, an entity cannot capitalize further costs and must apply the guidance in Section 3.3.1 on impairment to existing asset balances.

An entity must **cease** capitalizing internal-use software costs no later than the point at which a software project is substantially complete and ready for its intended use, that is, after all substantial testing is complete.

The following diagram illustrates the model:



BDO INSIGHTS: CHANGES TO ACCOUNTING FOR INTERNAL-USE SOFTWARE COSTS MAY REQUIRE NEW PROCESSES

ASU 2025-06 introduces new areas of judgment when accounting for software costs, including determining whether the probable-to-complete recognition threshold is met or significant development uncertainty exists (see Section 3.2.1.1). Entities may need to develop new or revise existing processes and controls.

3.2.1.1 Significant Development Uncertainty



FASB REFERENCES

ASC 350-40-25-12A

ASC 350-40-20: Performance Requirements

Performance requirements are what an entity needs the software to do (for example, functions or features).

The probable-to-complete recognition threshold is not met until significant uncertainty associated with the development activities of the software (or “**significant development uncertainty**”) has been resolved. If significant development uncertainty exists, the entity must defer capitalization of internal-use software costs.

In determining whether significant development uncertainty exists, an entity considers whether **either** of the following factors is present. If either factor is present, then the probable-to-complete recognition threshold is **not** met.

Novel or Unproven Software	Significant Performance Requirements
The software being developed has technological innovations or novel, unique or unproven functions or features and the uncertainty related to those technological innovations, functions, or features, if identified, has not been resolved through coding and testing .	The entity has not clearly defined the significant performance requirements of the software being developed or continues to substantially revise those requirements.

The term “performance requirements” is defined as “what an entity needs the software to do (for example, functions or features).” That definition is consistent with how performance requirements were described in the definition of the term “preliminary project stage” under the previous guidance.

BC45 of ASU 2025-06 states that the two factors above are consistent with the previous guidance that required entities to evaluate whether it is probable that a software project will be completed, and that the software will be used to perform the function intended, before capitalizing software costs. Therefore, if an entity considered other facts and circumstances in that evaluation under the previous guidance, it must continue to consider those facts and circumstances in applying the probable-to-complete recognition threshold under the new guidance.

BDO INSIGHTS: DEFINITION OF SIGNIFICANT PERFORMANCE REQUIREMENTS

The guidance includes the terms “significant development uncertainty” and “significant performance requirements” but does not define the term “significant” because it is widely used in U.S. GAAP and well understood in practice. While the two factors in Section 3.2.1.1 are helpful to determine whether significant development uncertainty exists, judgment is required to apply the definition of performance requirements and determine what constitutes significant performance requirements.

BDO INSIGHTS: ACCOUNTING FOR SOFTWARE COSTS UNDER ASC 350-40 VS ASC 985-20

The requirement in ASC 350-40 that uncertainty related to novel or unproven software be resolved through coding and testing is similar to the guidance in ASC 985-20 about resolution of uncertainties related to high-risk development issues. By including the concept of significant development uncertainty, the new guidance results in greater alignment between the outcomes under ASC 350-40 and ASC 985-20. As a result, the FASB expects¹ that more software development costs may be expensed under the revised guidance in ASC 350-40.

However, while the models in ASC 350-40 and ASC 985-20 are often expected to result in similar outcomes, differences might continue to exist between the two models because of certain nuanced differences in the guidance. For example, as discussed in Section 2.2.1, ASC 985-20 requires either a detail program design or a working model to support that there are no high-risk development issues and thus technological feasibility has been achieved. ASC 350-40 does not require similar support to conclude that high-risk development issues have been resolved. Thus, it is possible that capitalization under ASC 350-40 could begin earlier than under ASC 985-20.

Assessing whether significant development uncertainty exists may be straightforward for some software projects, such as the implementation and customization of an enterprise resource planning (ERP) system using a developed solution (see Example 3-1). For other software projects, the assessment may be more complex as illustrated in Example 3-2.

¹ BC57 of ASU 2025-06.



SAAS ENTITIES MAY CAPITALIZE FEWER COSTS

As discussed in BC57 of ASU 2025-06, the concept of significant development uncertainty could affect the financial reporting of SaaS entities that are developing novel or unproven software, by requiring most software costs to be expensed. Given that (a) a significant development uncertainty must be resolved through coding and testing for the capitalization criteria to be met and (b) capitalization ends after all substantial testing is complete, there may be limited time between the two milestones during which eligible costs would be capitalized.

If significant development uncertainty does not exist or if there was significant development uncertainty that has been resolved, an entity evaluates the capitalization criteria in Section 3.2.1 to determine when to begin capitalizing costs.

EXAMPLE 3-1: [ADAPTED FROM ASC 350-40-55-5 THROUGH 55-8] IMPLEMENTATION AND CUSTOMIZATION OF AN ERP SYSTEM

FACTS

- ▶ On February 1, 20X3, a professional services entity starts planning to implement an ERP system that will support its finance, human resources, accounting, and client relationship functions.
- ▶ After evaluating different solutions, management signed a contract with a third party on August 1, 20X3, to implement and customize a hybrid ERP solution that offers on-premises software and cloud computing services.
- ▶ The entity will make customization decisions throughout the development process to select from existing functions and features offered by the third party.

CONCLUSION

The entity starts capitalizing eligible software implementation and customization costs as of August 1, 20X3.

ANALYSIS

The entity evaluates whether the internal and external costs to implement and customize the ERP system meet the capitalization criteria under ASC 350-40-25-12 through 25-12A:

- ▶ The entity determines that significant development uncertainty does not exist as of August 1, 20X3 because:
 - It has identified the significant performance requirements and expects no substantial revisions, since customization involves selecting from existing functions and features.
 - The entity selected a pre-developed solution with no novel, unique, or unproven functions or features.
- ▶ The entity evaluates when to begin capitalizing software costs and determines:
 - Management authorized and committed to funding the software project on August 1, 20X3, by executing the contract with the third party.
 - Based on all relevant facts and circumstances, such as engaging an experienced third party to implement and customize the software, the software project is probable-to-complete, and the software will perform as intended.
- ▶ As a result, on August 1, 20X3, the entity begins capitalizing both:
 - Costs for implementation and customization of the on-premises software license
 - Costs for implementation of the cloud computing service features included in the hybrid solution.

EXAMPLE 3-2: [ADAPTED FROM ASC 350-40-55-13 THROUGH 55-17] DEVELOPING A NOVEL TECHNOLOGY**FACTS**

- ▶ On January 1, 20X1, a software development entity starts discussions to develop software with novel functionality.
- ▶ On February 1, 20X1, management completes its due diligence procedures, approves a budget for internal development, and assigns an internal team to start developing the novel software. At that time, the software still included novel functionality.
- ▶ On March 1, 20X3, the entity resolves the uncertainty related to the novel functionality through coding and testing. Management also confirms that it has identified significant performance requirements (the significant functions and features) for the software and does not expect substantial changes.
- ▶ On April 1, 20X3, the entity determines that all substantial testing is complete.

CONCLUSION

The entity starts capitalizing eligible software development costs on March 1, 20X3, and stops capitalizing costs on April 1, 20X3, when the software is substantially complete and ready for use.

ANALYSIS

The entity evaluates whether the internal and external costs to develop the software meet the capitalization criteria under ASC 350-40-25-12 through 25-12A:

- ▶ As of February 1, 20X1, the entity determines that significant development uncertainty exists because:
 - It has not yet identified the significant performance requirements.
 - The software includes novel functionality that has not yet been resolved through coding and testing.

Therefore, the entity cannot begin capitalizing software costs as of February 1, 20X1.

- ▶ As of March 1, 20X3, the entity determines that significant development uncertainty no longer exists because:
 - It has identified the significant performance requirements and does not expect substantial revisions.
 - It has resolved the uncertainty related to the novel functionality through coding and testing.
- ▶ On March 1, 20X3, the entity evaluates when to begin capitalizing software costs and determines:
 - The management authorized and committed to funding the software project on February 1, 20X1.
 - The software project is probable-to-complete, and the software will perform as intended based on all relevant facts and circumstances.
- ▶ As a result, on March 1, 20X3, the entity begins capitalizing eligible software costs.
- ▶ On April 1, 20X3, the entity determines that the software project is substantially complete and ready for its intended use because all substantial testing is complete. Therefore, the entity stops capitalizing software costs on April 1, 20X3.

EXAMPLE 3-3: [ADAPTED FROM ASC 350-40-55-9 THROUGH 55-12] DEVELOPING A MOBILE APPLICATION**FACTS**

- ▶ An entity is internally developing a mobile application (app) called X-Crowd, which will allow users to see how crowded a restaurant or store is based on user's real-time input. An internet connection is required to access the app.
- ▶ On February 1, 20X1, management approved funding for the internal development of the app. However, the entity has not yet identified specific functions and features to be included in the app.
- ▶ Through November 30, 20X1, the entity continues to develop the app's functions and features. This included gathering feedback from user groups on early versions and adjusting the development based on feedback.

- ▶ On December 1, 20X1, management determined that it has identified the significant performance requirements (the significant functions and features the app must have) and it does not expect substantial changes going forward.
- ▶ Throughout the development process, management concluded that the app does not involve technological innovations or novel, unique, or unproven functions or features.

CONCLUSION

The entity starts capitalizing eligible software development costs on December 1, 20X1.

ANALYSIS

The entity evaluates whether the internal and external costs to develop X-crowd meet the capitalization criteria under ASC 350-40-25-12 through 25-12A:

- ▶ As of February 1, 20X1, the entity determines that significant development uncertainty exists because:
 - While the app does not include technological innovations or novel, unique, or unproven functions or features, the entity has not yet identified the significant performance requirements.

Therefore, the entity cannot begin capitalizing software costs as of February 1, 20X1.

- ▶ The entity determines that significant development uncertainty is resolved as of December, 20X1 because:
 - It has identified the significant performance requirements and does not expect substantial revisions.
 - The app continues to lack technological innovations or novel, unique, or unproven functions or features.
- ▶ On December 1, 20X1, the entity evaluates when to begin capitalizing software costs and determines:
 - Management authorized and committed to funding the software project on February 1, 20X1.
 - The software project is probable-to-complete, and the app will perform as intended, based on all relevant facts and circumstances.
- ▶ As a result, on December 1, 20X1, the entity begins capitalizing eligible software costs.

3.2.2 Capitalizable Costs



FASB REFERENCES

ASC 350-40-25-4 through 25-5 and ASC 350-40-30-1 through 30-3

An entity can capitalize **only** the following costs of internal-use software after the capitalization criteria (see Section 3.2.1) are met:



External Direct Costs of Materials and Services

- ▶ External direct costs of materials and services consumed in developing or obtaining internal-use computer software. Examples include, but are not limited to:
 - Fees paid to third parties for services provided to develop the software.
 - Costs incurred to obtain software from third parties.
 - Travel expenses incurred by employees in their duties directly associated with developing software.

 Payroll and Related Costs	▶ 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 activities for which payroll costs are capitalized include, but are not limited to: • Design of chosen path, including software configuration and software interfaces. • Coding. • Installation of hardware. • Testing, including parallel processing phase.
 Interest Costs	▶ Interest costs incurred while developing internal-use computer software, capitalized in accordance with ASC 835-20, <i>Capitalization of Interest</i>. ▶ If the entity suspends substantially all activities related to the software developed or obtained for internal use, it must cease interest capitalization until activities are resumed.
 Data Conversion Costs	▶ Costs to develop or obtain software that allows for access to or conversion of old data by new systems. ▶ The process of data conversion from old to new systems may include: • Purging or cleansing of existing data. • Reconciling or balancing the data and the data in the old and new systems. • Creation of new or additional data. • Conversion of old data to the new system.

BDO INSIGHTS: AI USED TO DEVELOP SOFTWARE

An entity may use an AI tool as part of its software development process, for example, to write code. We believe the related costs represent external direct costs of materials and services that are eligible for capitalization and therefore must be included in the cost of internal-use software. Because an entity may use an AI tool for multiple purposes, it must apply judgment to determine which fees to capitalize as development costs.

An entity **cannot** capitalize the following as costs of internal-use software:

- ▶ General and administrative costs.
- ▶ Overhead costs.
- ▶ Data conversion costs, other than costs to develop or obtain software used in the conversion process capitalized as discussed above.
- ▶ Internal and external training costs, which are expensed as incurred.

BDO INSIGHTS: MANAGEMENT CONTROLS OVER DETERMINING COSTS TO CAPITALIZE

Entities that incur internal costs such as labor and related overhead to develop internal-use software must also develop processes and controls over determining which costs are capitalized. This area can be challenging in practice, especially if the entity develops multiple software projects at the same time. The entity should develop a process to:

- ▶ Document when the criteria for capitalization have been met (see Section 3.2.1).
- ▶ Determine which costs must be capitalized.
- ▶ Determine when the software project is complete and thus capitalization must end.
- ▶ Monitor and determine whether an impairment indicator exists (see Section 3.3.1).

In addition, the entity should establish appropriate management controls over each step of the process.

3.2.2.1 Nature of Costs Versus Timing of Incurrence**FASB REFERENCES**

ASC 350-40-55-4

An entity applies guidance on determining which costs are eligible for capitalization once the capitalization criteria are met based on the nature of the costs incurred, not the timing of their incurrence. For example, while an entity may incur some training costs after meeting the capitalization criteria and before the software project is substantially complete and ready for its intended use, it must nonetheless expense the training costs as incurred.

Additionally, software development activities that are performed simultaneously, for example, coding and testing, qualify for capitalization based on the nature of those costs if the capitalization criteria are met (see Section 3.2.1).

3.2.3 Unit of Account for the Capitalization Criteria in ASC 350-40

ASC 350-40 applies to internal-use software at the level of individual components or modules. Specifically, BC61 of ASU 2025-06 states that the unit of account in applying the capitalization criteria (see Section 3.2.1) is a software project. However, the guidance does not define the term “software project” or specify what constitutes a software project.

BDO INSIGHTS: DETERMINING THE UNIT OF ACCOUNT OF A SOFTWARE PROJECT

Determining what constitutes a software project may be unclear, particularly when an entity uses an agile method to develop software. For example, while one entity might view the unit of account to be an entire project, others might view the unit of account to be one phase of a project or even a single sprint. In the absence of specific guidance, an entity must apply judgment based on its facts and circumstances to determine the unit of account.

BC63 of ASU 2025-06 explains that the FASB did not provide additional guidance on determining the unit of account in ASU 2025-06 because that could have changed practice or limited the judgment that is allowed in the existing guidance. Therefore, entities will need to apply judgment to determine what constitutes a software project — a new product, a distinct module, or a major feature or function to an existing product.

3.2.3.1 Unit of Account for Software Embedded in Tangible Assets

Often software is embedded in hardware or other tangible assets, which raises questions about:

- ▶ The unit of account for an asset that incorporates both software components and tangible assets.
- ▶ Whether the software component is accounted for separately under ASC 350-40 or combined with the tangible asset in accordance with other U.S. GAAP, such as ASC 360-10.

BDO INSIGHTS: DETERMINING UNIT OF ACCOUNT FOR EMBEDDED SOFTWARE

When the FASB proposed amendments to the guidance on internal use software, it included a requirement that entities use a reasonable and consistent method to determine the unit of account for an asset that incorporates both software components and a tangible asset. However, based on feedback expressing concerns about the operability and auditability of the proposed language, ASU 2025-06 did not include this requirement.

In practice, when software is critical to or enhances the functionality of related property, plant and equipment (PPE), entities typically account for those software costs as one unit of account with the related PPE, an outcome the Board indicated it did not intend to change.

3.2.4 Multiple-Element Arrangements



FASB REFERENCES

ASC 350-40-30-4 and ASC 105-10-05-6

An entity may purchase internal-use software from a third party or may enter a hosting arrangement. If a contract to purchase internal-use software or enter a hosting arrangement includes multiple software and software-related elements, an entity must allocate the cost among the elements. The allocation is based on the relative standalone price of the elements in the contract and not necessarily separate prices stated within the contract for each element.

Examples of different elements that might be in the contract include: the license or hosting, training for the software, maintenance fees for routine maintenance work to be performed by the third party, data conversion costs, reengineering costs and rights to future upgrades and enhancements. See Section 1.2.3 for a discussion of hybrid cloud arrangements that include both an on-premise license and a cloud-based SaaS arrangement.

An entity accounts for the elements based on the nature of the cost incurred. For example, an element in the scope of ASC 350-40 is accounted for in accordance with guidance in this chapter.

When identifying elements in a contract, the guidance need not be applied to immaterial items.²

BDO INSIGHTS: RELATIVE STANDALONE PRICE

The guidance does not define the term “relative standalone price.” In practice, relative standalone price is considered synonymous with the term “relative standalone selling price” as described in ASC 606, which represents the price at which an entity would purchase an element of a contract separately (see section 5.3 of BDO’s Blueprint, [Revenue Recognition Under ASC 606](#)). Determining the relative standalone price of each element in a contract requires the use of judgment, based on the facts and circumstances. An entity must consider all relevant information in estimating the relative standalone price of an element.

² BC17 of ASU 2018-15.

3.2.5 In-Licensing of Internal-Use Software



FASB REFERENCES

ASC 350-40-25-17, ASC 350-30-25-1, ASC 350-30-30-1, and ASC 805-50-30-1 through 30-4

Entities often license internal-use software from third parties. An entity accounts for the in-licensing of a software license within the scope of ASC 350-40 as the:

- ▶ Acquisition of an intangible asset.
- ▶ Incurrence of a liability, to the extent that all or a portion of the software licensing fees are not paid on or before the acquisition date of the license by the licensee.

An entity recognizes and measures the acquired intangible asset in accordance with ASC 350-30-25-1 and ASC 350-30-30-1, respectively. ASC 350-30-30-1 requires an entity to apply ASC 805-50-30-1 through 30-4 to measure the initial acquisition of assets that are not businesses. That guidance requires measuring an asset acquisition at cost, including any related transaction costs (see Appendix C of BDO's Blueprint, [Business Combinations Under ASC 805](#)).

BDO INSIGHTS: LICENSE FEES PAID OVER TIME

Often, the entire license fee related to a software license is due at inception of the license arrangement. However, in other cases the license fee may be paid over time. Once an entity obtains the right to use a software license, all fixed fees³ are recognized in the cost of that asset, with an offsetting liability reflecting any fees not yet paid.

In addition, the license fee may include variable payments based on the entity's use of the software. Neither ASC 350-40 nor ASC 805-50 provides guidance on recognizing consideration that is contingent on future events. As further discussed in Appendix C of BDO's Blueprint, [Business Combinations Under ASC 805](#), assuming the contingent consideration is outside the scope of ASC 815, *Derivatives and Hedging*, we believe that variable fees should be recognized as an additional cost of the software license either when they become probable and estimable (under the loss contingency model in ASC 450, *Contingencies*) or when the contingency is resolved (based on superseded guidance from paragraph 27 of Statement of Financial Accounting Standards No. 141, *Business Combinations*). Either approach is acceptable, but an entity must make a policy election that is consistently applied.

3.3 SUBSEQUENT MEASUREMENT



FASB REFERENCES

ASC 350-40-35-1 through 35-10

This section provides guidance on considerations for:

- ▶ Impairment of capitalized internal-use software
- ▶ Amortization of capitalized software
- ▶ Internal-Use software that is subsequently marketed.

³ In determining which fees are fixed, an entity considers the term of the contract including any termination provisions that can be exercised without significant penalty.

3.3.1 Impairment of Capitalized Internal-Use Software



FASB REFERENCES

ASC 350-40-35-1 through 35-3 and ASC 360-10-35-47 through 35-49

An entity recognizes and measures impairment of capitalized software in accordance with ASC 360-10-35, which applies to property, plant, and equipment. That guidance requires an entity to group assets at the lowest level for which there are identifiable cash flows that are largely independent of the cash flows of other groups of assets. This guidance applies, for example, when one of the following events or changes in circumstances occurs related to computer software being developed or currently in use indicating that the carrying amount may not be recoverable:

- ▶ Internal-use software is not expected to provide substantive service potential.
- ▶ A significant change occurs in the extent or way the software is used or is expected to be used.
- ▶ A significant change is made or will be made to the software program.
- ▶ Costs of developing or modifying internal-use computer software significantly exceed the amount originally expected to develop or modify the software.

Additionally, ASC 360-10-35-47 through 35-49 require that the asset be accounted for as abandoned when it ceases to be used.

3.3.1.1 Software Development Activities Trigger Reconsideration of Remaining Useful Lives of Existing Software



FASB REFERENCES

ASC 350-40-25-15

New software development activities may trigger a reconsideration of remaining useful lives of existing software that is to be replaced. When an entity replaces existing software with new software, any remaining unamortized costs of the old software are expensed when the new software is ready for its intended use.

3.3.1.2 Internal-Use Software Development Asset: Capitalization Criteria Are No Longer Met



FASB REFERENCES

ASC 350-40-35-1 through 35-3

If the capitalization criteria in Section 3.2.1 are no longer met for software being developed, an entity reports the software asset at the **lower** of the carrying amount or fair value, if any, less costs to sell regardless of whether the carrying amount of an asset group containing that uncompleted software is deemed to be recoverable. The impairment assessment for uncompleted software is performed at the module or component level.

The guidance includes a rebuttable presumption that the uncompleted software has a fair value of zero.

Indications that the capitalization criteria in Section 3.2.1 are no longer met include the following:

- ▶ A lack of expenditures budgeted or incurred for the project.
- ▶ Programming difficulties that cannot be resolved on a timely basis.
- ▶ Significant cost overruns.

- ▶ Information has been obtained indicating that the costs of internally developed software will significantly exceed the cost of comparable third-party software or software products, so that management intends to obtain the third-party software or software products instead of completing the internally developed software.
- ▶ Technologies are introduced in the marketplace, so that management intends to obtain the third-party software or software products instead of completing the internally developed software.
- ▶ Business segment or unit to which the software relates is unprofitable or has been or will be discontinued.

BDO INSIGHTS: RARE THAT FAIR VALUE IS GREATER THAN ZERO

As noted above, ASC 350-40 includes a rebuttable presumption that the fair value of any uncompleted software for which the capitalization criteria are no longer met is zero. Therefore, we believe it will be rare that an entity concludes that the capitalization criteria are no longer met for a software project, but the uncompleted project has a substantial fair value.

Concluding that the fair value of an uncompleted software project for which the capitalization criteria are no longer met is greater than zero requires the application of professional judgment based on the facts and circumstances.

3.3.2 Amortization of Capitalized Internal-Use Software



FASB REFERENCES

ASC 350-40-35-4 through 35-6

ASC 350-40-20: Useful Life

The period over which an asset is expected to contribute directly or indirectly to future cash flows.

An entity amortizes the costs of software developed or obtained for internal use on a straight-line basis unless another systematic and rational basis is more representative of the software's use. Amortization begins when the software is ready for its intended use (that is, after all substantial testing is complete), regardless of when the software is placed in service.

Given the history of rapid changes in technology, software often has a relatively short useful life. An entity must consider the effects of **all** the following in determining and periodically reassessing the estimated useful life over which the costs incurred for internal-use computer software will be amortized:

- ▶ Obsolescence
- ▶ Technology
- ▶ Competition
- ▶ Other economic factors
- ▶ Rapid changes that may occur in the development of software products, software operating systems, or computer hardware and whether management intends to replace any technologically inferior software or hardware.

3.3.2.1 Unit of Account for Amortization of Capitalized Internal-Use Software



FASB REFERENCES

ASC 350-40-35-6 and ASC 350-40-15-2

An entity evaluates the commencement of amortization at the module or component level. For each module or component of a software project, amortization begins when the computer software is ready for its intended use, regardless of whether the software will be placed in service in planned stages that may extend beyond a reporting period. Software is ready for its intended use after all substantial testing is complete. If the functionality of a module is entirely dependent on the completion of other modules, amortization of that module begins when both that module and the other modules upon which it is functionally dependent are ready for their intended use.

BDO INSIGHTS: DETERMINING WHAT CONSTITUTES A MODULE OR COMPONENT

The FASB did not provide additional insight in determining what constitutes a software system or a distinct module or component. Therefore, making this assessment requires the application of professional judgment based on the facts and circumstances.

EXAMPLE 3-4: [ADAPTED FROM ASC 350-40-15-2(D)] MODULE OR COMPONENT OF AN ACCOUNTING SOFTWARE SYSTEM

FACTS

An entity develops an accounting software system containing three elements: a general ledger, an accounts payable subledger, and an accounts receivable subledger.

CONCLUSION

Each of the three elements is a module of the entire accounting software system.

ANALYSIS

The entity determines that each of the three elements is a module of the entire accounting software system because each is capable of being used and provides value on its own.

3.3.3 Internal-Use Software Subsequently Marketed



FASB REFERENCES

ASC 350-40-35-7 through 35-10

An entity that had no plan to externally market software to other parties may subsequently decide to license or sell the software. In that case, the entity would have applied guidance in ASC 350-40 to the development of the software and may have capitalized some development costs.

If an entity subsequently decides to market internal-use software **after** its development is complete, proceeds received from licensing the software, net of direct incremental costs of marketing, reduce the carrying amount of that software. Examples of direct incremental costs of marketing include commissions, software reproduction costs, warranty and service obligations, and installation costs. An entity cannot recognize any profit **until** the total of (a) the net proceeds from licenses and (b) amortization reduces the software's carrying amount to zero.

Subsequent proceeds are recognized as **either**:

- ▶ Revenue in accordance with ASC 606 if the contract is with a customer (see BDO's Blueprint, [Revenue Recognition Under ASC 606](#)).
- ▶ A gain in accordance with ASC 610-20, *Gains and Losses from the Derecognition of Nonfinancial Assets*, if the contract is not with a customer (see Appendix A of BDO's Blueprint, [Revenue Recognition Under ASC 606](#)).

If an entity subsequently decides to market an internal-use software to others **during** its development, the entity must apply the guidance ASC 985-20:

- ▶ Amounts previously capitalized under ASC 350-40 are evaluated under a net realizable value test at each balance sheet date (see Section 2.4.2).
- ▶ Capitalized software costs are amortized (see Section 2.4.1).

A pattern of deciding to market internal-use software during its development creates a rebuttable presumption that any software developed by that entity is intended for sale, lease or other marketing, and therefore is subject to ASC 985-20.

3.4 UPGRADES AND ENHANCEMENTS



FASB REFERENCES

ASC 350-40-25-17A through 25-17E

Upgrades and enhancements are defined as modifications to existing internal-use software that result in additional functionality or the ability to perform additional tasks – that is, modifications that enable the software to perform tasks that it was previously incapable of performing.

Upgrades and enhancements usually require new software specifications and may also require a change to all or part of the existing software specifications. Accounting for the costs incurred to develop upgrades and enhancements to internal-use software follows the same model as other internal-use software costs **if** it is probable that those expenditures will result in additional functionality:

- ▶ Internal and external costs incurred for upgrades and enhancements are expensed or capitalized in accordance with the guidance in Section 3.2.
- ▶ Internal and external costs incurred for maintenance are expensed as incurred.
- ▶ If an entity cannot separate internal costs on a reasonably cost-effective basis between maintenance and relatively minor upgrades and enhancements, it must expense such costs as incurred.
- ▶ External costs incurred under agreements related to specified upgrades and enhancements are expensed or capitalized in accordance with the guidance in Section 3.2. If maintenance is combined with specified upgrades and enhancements in a single contract, the cost is allocated between the elements as discussed in Section 3.2.3 and the maintenance costs are expensed over the contract period. However, external costs related to maintenance, unspecified upgrades and enhancements and costs under agreements that combine the costs of maintenance and unspecified upgrades and enhancements are recognized in expense over the contract period on a straight-line basis unless another systematic and rational basis is more representative of the services received.

BDO INSIGHTS: UPGRADES AND ENHANCEMENTS

Determining whether upgrades and enhancements result in additional functionality to existing software requires the exercise of judgment based on facts and circumstances. Examples of enhancements that result in additional functionality to existing software include modifications that allow the software to be compatible with additional hardware.

BDO INSIGHTS: “ENHANCEMENT” IN ASC 985-20 VERSUS ASC 350-40

There are differences between the definitions of “enhancement” under ASC 985-20 and ASC 350-40. Under ASC 985-20, enhancements are defined as improvements that extend the useful life of the software **only**. However, under ASC 350-40, projects that result in additional functionality to existing software without extending the useful life of that software are also considered enhancements subject to capitalization.

BDO INSIGHTS: CAPITALIZATION THRESHOLDS

ASC 350-40 does not allow a capitalization threshold specifying the minimum amount of costs that must be incurred before capitalizing those costs. However, in practice entities often establish such a threshold for administrative convenience when expensing such amounts would not be material to the financial statements as a whole. If an entity elects to establish a capitalization threshold related to software enhancements, it must apply judgment to determine an appropriate capitalization threshold. In addition, the entity must establish processes and controls to evaluate the impact of such policy on the financial statements.

3.5 IMPLEMENTATION COSTS OF CCA THAT IS A SERVICE CONTRACT

**FASB REFERENCES**

ASC 350-40-25-18

ASC 350-40-20: Hosting Arrangement

In connection with accessing and using software products, an arrangement in which the customer of the software does not currently have possession of the software; rather, the customer accesses and uses the software on an as-needed basis.

A hosting or cloud computing arrangement (CCA) refers to an arrangement whereby a customer remotely accesses software hosted on the vendor’s or a third-party’s cloud platform servers.

While ASC 340-50 applies to costs incurred to develop internal-use software, aspects of the guidance also apply to implementation costs incurred to implement a CCA that is a service contract.

- ▶ If a CCA includes a license to internal-use software, the customer accounts for the software license in accordance with the guidance on internal-use software in ASC 350-40 (see Sections 3.2, 3.3 and 3.4).
- ▶ If a CCA does **not** include a software license, the customer accounts for the arrangement as a service contract and applies the specific guidance in ASC 350-40.

- See Sections 3.5.1, 3.5.2 and 3.5.3 on accounting for related implementation, setup, and other upfront costs (collectively referred to as “implementation costs”).
- See Section 1.4 for guidance on determining whether a CCA includes a software license.

The following discussion in this section relates to a CCA that is a service contract — that is, the CCA does **not** include a software license.

3.5.1 Implementation Costs of CCA that is a Service Contract: Recognition and Initial Measurement



FASB REFERENCES

ASC 350-40-25-18 and ASC 350-40-30-5

An entity applies guidance on internal-use software (see Section 3.2) as though the hosting arrangement is an internal-use computer software project to determine whether and when the implementation costs of a CCA that is a service contract are capitalized.



FASB UPDATE — AMENDMENTS TO THE INTERNAL-USE SOFTWARE GUIDANCE IN ASC 350-40

The FASB issued ASU 2025-06 in September 2025 to amend the internal-use software guidance in ASC 350-40. The amendments do **not** change the existing guidance in ASC 350-40 on accounting for implementation costs of CCA. Nevertheless, the changes to the capitalization criteria for internal-use software development costs, which also apply to implementation costs of a CCA, result in closer alignment between the recognition requirements for software costs that are monetized via a CCA (for example, SaaS) and the requirements for software costs that are sold via an on-premises license.

This publication assumes that an entity has adopted ASU 2025-06. See Chapter 5 for transition guidance.

3.5.2 Impairment of Capitalized Implementation Costs Related to CCA that is a Service Contract



FASB REFERENCES

ASC 350-40-35-11 through 35-12 and ASC 360-10-35-47 through 35-49

An entity recognizes and measures impairment for a CCA that is a service contract in accordance with ASC 360-10-35 as if the capitalized implementation costs were a long-lived asset. That guidance requires assets to be grouped at the lowest level for which there are identifiable cash flows that are largely independent of the cash flows of other groups of assets.

The impairment guidance applies when events or changes in circumstances occur related to the CCA indicating that the carrying amount of the related implementation costs may not be recoverable. Examples of such events or changes in circumstances include but are not limited to:

- ▶ The hosting arrangement is not expected to provide substantive service potential.
- ▶ A significant change occurs in the extent or way the hosting arrangement is used or is expected to be used.
- ▶ A significant change is made or will be made to the hosting arrangement.

Additionally, ASC 360 requires an entity to account for an asset as abandoned when it ceases to be used. Accordingly, an entity evaluates the implementation costs related to each module or component separately to determine when it ceases to be used. That is, the unit of account for evaluating abandonment is each module or component in a CCA.

3.5.3 Amortization of Capitalized Implementation Costs of CCA



FASB REFERENCES

ASC 350-40-35-13 through 35-17

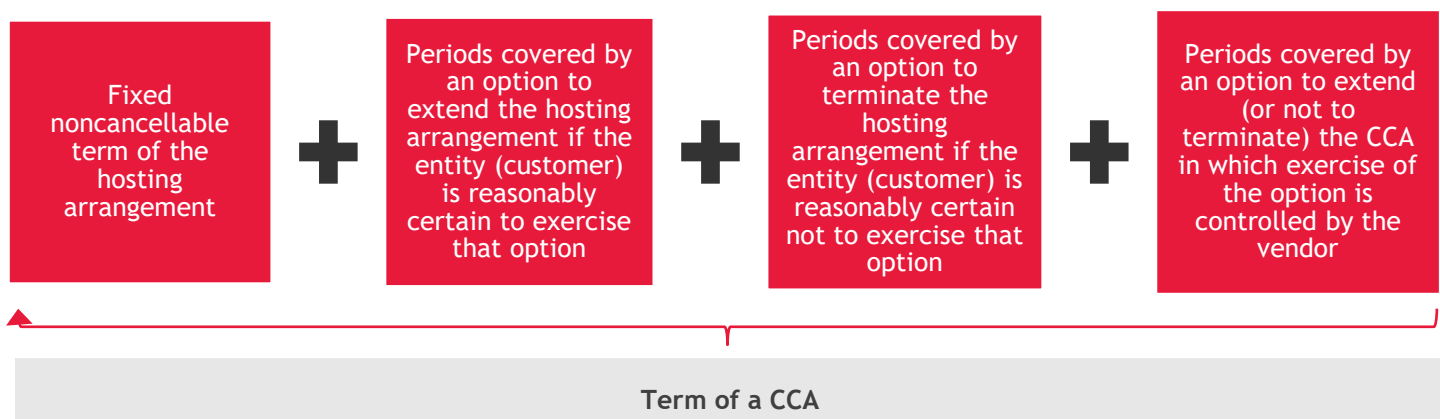
An entity amortizes the implementation costs capitalized in accordance with Section 3.5.1 over the CCA's term on a straight-line basis unless another systematic and rational basis is more representative of the pattern in which the entity expects to benefit from access to the hosted software. For this purpose, the right to access the hosted software is not affected by the extent to which the entity uses, or the expectations about the entity's use of, the hosted software (for example, how many transactions the entity processes or expects to process or how many users access or are expected to access the hosted software).

The unit of account amortizing capitalized implementation costs is each module or component in a CCA. For each module or component of a CCA, an entity begins amortizing the capitalized implementation costs related to the CCA when the module or component of the CCA is ready for its intended use, regardless of whether the overall CCA will be placed in service in planned stages that may extend beyond a reporting period.

A CCA (or a module or component of a CCA) is ready for its intended use after all substantial testing is complete. If the functionality of a module or component is entirely dependent on completing other modules or components, an entity begins amortizing the capitalized implementation costs related to that module or component when both that module or component and the other modules or components upon which it is functionally dependent are ready for their intended use.

3.5.3.1 Term of a CCA

As previously discussed, an entity amortizes capitalized implementation costs over the CCA's term. For this purpose, the term of a CCA includes both the initial term and specific additional periods as illustrated in the following diagram:



3.5.3.2 Changes in Term

An entity is required to periodically reassess the estimated term of the arrangement and account for any change in the estimated term as a change in estimate in accordance with ASC 250 (see BDO's Publication, [Accounting Changes and Error Corrections](#)).

An entity must consider the effects of **all** the following when determining (reassessing) the CCA’s term:

- ▶ Obsolescence
- ▶ Technology
- ▶ Competition
- ▶ Other economic factors
- ▶ Rapid changes are occurring in the development of hosting arrangements or hosted software
- ▶ Significant implementation costs are expected to have significant economic value for the entity when the option to extend or terminate the hosting arrangement becomes exercisable

3.6 WEBSITE DEVELOPMENT COSTS

 FASB REFERENCES

ASC 350-40-25-17F through 25-17J and ASC 720-35-25-1

An entity applies ASC 350-40 to account for website development costs as follows:

 Fees Incurred for Website Hosting	▶ Website hosting involves the payment of a specified periodic fee to an internet service provider in exchange for hosting the website on its servers connected to the internet. ▶ Fees incurred for website hosting expensed over the period of benefit.
 Costs to Input Content	▶ Costs to input content into a website are expensed as incurred. ▶ Accounting for website content involves issues that also apply to other forms of content or information that are not unique to websites. Content refers to information included on the website, which may be textual or graphical in nature. However, the specific graphics described below are excluded from content and may be eligible for capitalization: • Graphics that are a component of software: The costs of developing initial graphics are evaluated for capitalization under ASC 350-40 for internal-use software and ASC 985-20 for software marketed externally. • Graphics that involve the overall design of the web page (use of borders, background and text colors, fonts, frames, buttons, and so forth) that affect the look and feel of the web page and generally remain consistent regardless of changes made to the content: The costs to develop such graphics are capitalized to the extent spent directly on the project if the criteria for capitalization in ASC 350-40 are met. ▶ Content may be coded directly into the web pages, or it may reside in separate databases that are integrated into or accessed from the web page with software. ▶ Examples of content include the following: • Articles • Product photos • Maps • Stock quotes • Charts

 Costs to Register the Website with Search Engines	► Costs to register the website with internet search engines represent advertising costs and are expensed as incurred.
 Costs to Obtain and Register Internet Domain	► Costs to obtain and register an internet domain are evaluated for capitalization as intangible assets under ASC 350-30-25.



FASB UPDATE: AMENDMENTS TO THE INTERNAL-USE SOFTWARE GUIDANCE IN ASC 350-40

The FASB issued ASU 2025-06 in September 2025 to amend the internal-use software guidance in ASC 350-40. The amendments modernize the internal-use software guidance to accommodate the evolution in software development method from a linear development method to a nonlinear development method. Among other changes, ASU 2025-06 supersedes the previous guidance for website development costs in ASC 350-50 and incorporates the recognition requirements for website-specific development costs into ASC 350-40, aligning it with the new guidance for internal-use software.

When the previous guidance for website development costs in ASC 350-50 was issued, websites were typically used by entities to promote products, replace manual processes or services and sell products. Since then, technology has rapidly evolved, and, currently, websites are most often used as an interface to access underlying software, the development of which is usually accounted for under ASC 350-40. That resulted in infrequent application of the previous guidance for website development costs or immaterial costs to develop a website separately from the underlying software. Therefore, the FASB superseded the previous guidance for website development costs in ASC 350-50 and incorporated relevant and incremental guidance that is unique to website-specific development costs into ASC 350-40.

This publication assumes that an entity has adopted ASU 2025-06. See Chapter 5 for transition guidance.

EXAMPLE 3-5: [ADAPTED FROM ASC 350-40-55-18 THROUGH 55-21] DEVELOPMENT OF A WEBSITE

FACTS

- On June 15, 20X1 an animal rescue organization starts discussions to develop a website that will share information with users, including hours of operation, contact details, animals available for adoption, and standard adoption procedures.
- On August 1, 20X1, after researching different website developers and performing due diligence, management executes a contract with a third party to develop the website. The third party is an established website developer and offers templates that the organization can use to create its website.
- In addition to website development fees paid to the third party, the organization incurs costs to:
 - Obtain and register an internet domain
 - Input content into the website
 - Develop initial graphics for the website
 - Register the website with internet search engines
 - Pay ongoing website hosting fees.

CONCLUSION

The organization begins capitalizing eligible website development costs on August 1, 20X1, when the capitalization criteria are met. Specific costs (such as content input and search engine registration) are expensed as incurred, while others (such as development fees and initial graphics) are evaluated for capitalization under ASC 350-40.

ANALYSIS

The organization assesses whether the internal and external costs to develop the website meet the capitalization requirements in paragraphs 350-40-25-12 through 25-12A, as follows:

- ▶ As of August 1, 20X1, the organization determines that significant development uncertainty does not exist because:
 - It has identified the significant performance requirements and does not expect substantial revisions – the website will be created from existing templates.
 - The website does not include technological innovations or novel, unique, or unproven functionality.
- ▶ The organization evaluates when to begin capitalizing costs and determines:
 - Management authorized and committed to funding the project on August 1, 20X1, when it executed the contract with the third party.
 - Based on all relevant facts and circumstances (including engaging an experienced third-party developer), it is probable that the project will be completed and the website will perform as intended.
- ▶ As a result, on August 1, 20X1, the organization begins capitalizing eligible costs and evaluates each cost type as follows:
 - Fees paid to the third party for website development: evaluated for capitalization under ASC 350-40-30-1.
 - Costs to obtain and register the internet domain: evaluated for capitalization under ASC 350-40-25-17J.
 - Costs to input content into the website: expensed as incurred under ASC 350-40-25-17G.
 - Costs to develop initial graphics: evaluated for capitalization under ASC 350-40-25-17H.
 - Costs to register the website with search engines: expensed as incurred under ASC 350-40-25-17I.
 - Ongoing website hosting fees: expensed over the period of benefit under ASC 350-40-25-17F.

Chapter 4: Presentation and Disclosures



4.1 OVERVIEW

ASC 350-40 and ASC 985-20 provide limited guidance on presentation and disclosure requirements for software costs related to software to be sold, leased, or marketed, internal-use software, and cloud computing arrangements.

Presentation guidance addresses the classification of capitalized software costs on the balance sheet, the presentation of amortization in the income statement, and the classification of cash flows, with additional considerations for purchased software and implementation costs in cloud computing arrangements (see Sections 4.2, 4.2.1, and 4.2.2).

Disclosure requirements include unamortized software costs and amounts expensed. Recent amendments under ASU 2025-06 require entities to align internal-use software disclosures with those for property, plant, and equipment (see Sections 4.3, 4.3.1, and 4.3.2).

If ASC 350-40 or ASC 985-20 does not provide specific presentation or disclosure requirements, an entity applies the general financial statement presentation principles and other U.S. GAAP, as appropriate.

4.2 PRESENTATION

The table below summarizes the presentation requirements for software to be sold, leased, or marketed, Internal-use software and cloud computing arrangements:

	SOFTWARE TO BE SOLD, LEASED, OR MARKETED	INTERNAL-USE SOFTWARE	CLOUD COMPUTING ARRANGEMENTS
Balance Sheet	Present capitalized software costs with useful life > 1 year as other assets (amortizable intangible asset).	No specific requirements.	Present capitalized implementation costs in the same line that a prepayment of fees for the associated hosting arrangement is presented.

	SOFTWARE TO BE SOLD, LEASED, OR MARKETED	INTERNAL-USE SOFTWARE	CLOUD COMPUTING ARRANGEMENTS
Income Statement	Present amortization of capitalized software costs in cost of sales or a similar expense category.	Present amortization in operating expenses (for example as depreciation and amortization).	Present amortization in operating expenses in the same line as the related hosting fees.
Cash Flow Statement (see BDO's Blueprint, Statement of Cash Flows Under ASC 230)	Classify cash flows related to capitalized software costs as operating activities .	Classify cash flows for capitalized costs as investing activities .	Classify cash flows for capitalized implementation costs as operating activities .

4.2.1 Software to be Sold, Leased, or Marketed



FASB REFERENCES

ASC 985-20-45-1 through 45-3

Capitalized development costs related to software intended to be sold, leased, or marketed, whether internally developed or purchased, are typically presented as amortizable intangible assets. If the software is purchased for resale, classification as inventory may be appropriate.

While ASC 350-40 does not govern the recognition and measurement of costs associated with software to be sold, leased, or marketed, its presentation and disclosure requirements apply. Accordingly, an entity must follow the relevant guidance in ASC 350-40 when presenting and disclosing these costs.

An entity presents amortization of capitalized software costs within cost of sales or a similar expense line because the expense relates directly to revenue from marketed software products.

BDO INSIGHTS: PRESENTATION OF PURCHASED SOFTWARE COSTS

The guidance does not prescribe whether software purchased for resale or to incorporate into another product must be presented separately or combined with other capitalized costs. We believe presentation depends on timing and intended use:

- ▶ If acquired after technological feasibility is established for the related product, generally present it together with other capitalized production costs.
- ▶ If acquired before technological feasibility is established and it has an alternative future use, classify it based on that use (for example, inventory).
- ▶ If feasibility is later achieved, reclassify the cost so it is presented with related production costs.
- ▶ If feasibility is never achieved, continue presenting it based on its alternative use to avoid misleading classification.

Determining the appropriate presentation of purchased software costs requires the application of professional judgment based on the facts and circumstances.

See Example 4-1 for an illustration of the presentation of purchased software costs.

EXAMPLE 4-1: [ADAPTED FROM ASC 985-20-55-14] PRESENTATION OF SOFTWARE PURCHASED BEFORE TECHNOLOGICAL FEASIBILITY IS ESTABLISHED**FACTS**

Assume the same facts as Example 2-1, as follows:

- ▶ An entity purchases software for \$100,000 before technological feasibility of the software has been established.
- ▶ The entity can resell the purchased software for \$75,000.

CONCLUSION

The entity accounts for the cost of purchased software as follows:

- ▶ \$25,000 is expensed as research and development.
- ▶ \$75,000 is capitalized as an intangible asset.

ANALYSIS

The entity determines that the software has an alternative future use at the purchase date because it can resell the software for \$75,000. Therefore, the entity capitalizes \$75,000 as an intangible asset.

For presentation purposes:

- ▶ If the software product reaches technological feasibility, \$75,000 is included in the cost of the software product and amortized. The amortization expense is presented in cost of sales.
- ▶ If the technological feasibility is never established, the \$75,000 is presented as inventory until the purchased software is sold or impaired.

4.2.2 Internal-Use Software and CCA

4.2.2.1 Internal-Use Software

**FASB REFERENCES**

ASC 350-40-25-17

Entities often license internal-use software from third parties. ASC 350-40 does not prescribe presentation requirements. Therefore, an entity must elect an accounting policy for presenting capitalized implementation costs on the balance sheet and apply it consistently. Acceptable approaches include but are not limited to presenting internal-use software as an intangible asset in same manner as licensed software or as a component of property, plant, and equipment. If any portion of the licensing fee remains unpaid at the acquisition date, a corresponding liability is recorded for the unpaid amount.

Amortization of internal-use software costs is presented in the income statement within operating expenses, commonly with depreciation and amortization. On the cash flow statement, cash outflows for capitalized internal-use software costs are presented as investing activities, consistent with other capital expenditures.

4.2.2.2 Cloud Computing Arrangement



FASB REFERENCES

ASC 350-40-45-1 through 45-3

An entity presents implementation costs for cloud computing arrangements consistently with the related hosting fees. On the balance sheet, capitalized implementation costs are presented in the same line as any prepayment for hosting fees. In the income statement, an entity reports amortization of capitalized implementation costs in the same line as the related hosting service expenses. In the statement of cash flows, implementation costs are classified in the same manner as cash flows for hosting fees.

BDO INSIGHTS: PRESENTATION OF CLOUD COMPUTING COST OR HOSTING FEES

Cloud computing arrangements or hosting services, such as SaaS, are accounted for as service contracts when the customer does not take possession of the software. In these cases, the fees paid for hosting services are presented as operating expenses in the income statement. On the balance sheet, any prepayments for hosting fees are recorded as prepaid assets (not intangible assets). In the statement of cash flows, payments for hosting services are classified as operating activities, consistent with other service arrangements.

If the CCA includes a software license, the license-related costs follow the guidance for internal-use software or software intended for sale, which differ from the service contract presentation.

4.3 DISCLOSURES

This section outlines disclosure requirements for software costs, including both software to be sold or licensed and internal-use software, as well as hosting arrangements that are service contracts. It also summarizes disclosure requirements introduced by recent amendments under ASU 2025-06.

- ▶ Section 4.3.1: Disclosure Requirements for Software to Be Sold, Leased, or Marketed
- ▶ Section 4.3.2: Disclosure Requirements for Internal-Use Software and Cloud Computing Arrangements

4.3.1 Software to be Sold, Leased, or Marketed



FASB REFERENCES

ASC 985-20-50-1 through 50-2, ASC 275-10-50-8, ASC 985-20-55-23 through 55-27, and ASC 985-20-55-29

ASC 985-20 requires an entity to disclose **both** of the following in the financial statements:

- ▶ Unamortized computer software costs included in each balance sheet presented.
- ▶ The total expense in each income statement presented for **both** of the following:
 - Amortization of capitalized computer software costs.
 - Write-downs to net realizable value.

Amortization and write-downs may be combined and disclosed as a single total.

ASC 350-30 requires an entity to apply the disclosure requirements in ASC 350-30-50 to capitalized software costs. Additionally, ASC 730-10 requires disclosure of the total research and development expense for each period presented in the income statement, including costs incurred for a computer software product to be sold, leased, or marketed.

ASC 275, *Risks and Uncertainties*, requires an entity to disclose information when information is known before issuing the financial statements that indicates that a material change in these estimates is reasonably possible in the near term due to future events. Thus, an entity must disclose events or circumstances that may result in a change in the expected useful life of capitalized software costs that may have a material effect on the financial statements. Example 4-2 illustrates the disclosure of the reasonable possibility that the carrying amount of capitalized costs may decrease in the near term because of changes in estimates.

If the amortization policy is disclosed elsewhere in the notes, it does not need to be repeated. Disclosure of factors that make the estimate sensitive to change, such as competitive pressures, is encouraged but not required by ASC 275.

EXAMPLE 4-2: [ADAPTED FROM ASC 985-20-55-23 THROUGH 55-27] DISCLOSURE OF RISKS AND UNCERTAINTIES RELATED TO CAPITALIZED SOFTWARE COSTS

Software, Inc. develops and markets computer programs. In 20X3, the company acquired a software entity and allocated a significant portion of the purchase price to Product A, its most profitable and widely marketed software program. Product A's carrying amount is \$5 million. Management expects Product A and its derivatives to remain among its most significant products for the next several years. However, a competitor recently introduced a product that directly competes with Product A.

Software, Inc. amortizes capitalized software costs using the greater of:

- ▶ The ratio of current gross revenues for a product to the total of current and anticipated future gross revenues for that product, and
- ▶ The straight-line method over the remaining estimated economic life of the product, including the current reporting period.

For 20X4, amortization will equal 20% of the beginning-of-year capitalized amount and will represent a significant component of cost of sales.

The computer software industry segment in which Software, Inc. operates is highly competitive and subject to rapid technological change. Customer purchasing decisions are influenced by product reviews and advertising, and innovative products introduced by competitors have historically reduced sales volumes of existing products quickly and significantly. While management believes its estimates of future gross revenues and the economic life of Product A are reasonable, new, competitive products could materially affect these estimates in the near term. As a result, periodic amortization could increase significantly, and the carrying amount of capitalized software costs for Product A (\$5 million) may be reduced materially.

Disclosure:

Software, Inc.'s policy is to amortize capitalized software costs by the greater of the following:

- (a) The ratio that current gross revenues of a product bear to the total of current and anticipated future gross revenues for that product.
- (b) The straight-line method over the remaining estimated economic life of the product including the current reporting period.

It is reasonably possible that estimates of anticipated future gross revenues, remaining estimated economic life of the product, or both will be reduced significantly in the near term due to competitive pressures. As a result, the carrying amount of capitalized software costs for Product A (\$5 million) may be reduced materially in the near term.

4.3.2 Internal-Use Software and CCA



FASB REFERENCES

ASC 350-40-50-1 through 50-3

ASC 350-40 does not prescribe specific disclosure requirements for internal-use software costs beyond those required by other authoritative literature. Entities must consider disclosure requirements referenced in other U.S. GAAP, including:

- ▶ ASC 275, *Risks and Uncertainties*
- ▶ ASC 730-10, *Research and Development*
- ▶ ASC 235, *Notes to Financial Statements*
- ▶ ASC 360-10, *Property, Plant, and Equipment*

For hosting arrangements that are service contracts, ASC 350-40 requires disclosure of the nature of the arrangement. In addition, entities must apply disclosure requirements in ASC 360-10, as if capitalized implementation costs were a separate major class of depreciable asset.



FASB UPDATE: DISAGGREGATION OF INCOME STATEMENT EXPENSES

In November 2024, the FASB issued ASU 2024-03, *Income Statement – Reporting Comprehensive Income – Expense Disaggregation Disclosures (Subtopic 220-40): Disaggregation of Income Statement Expense*, which requires that an entity include existing required disclosures of the amortization of capitalized implementation costs of hosting arrangements that are service contracts in a newly required tabular disclosure. ASU 2024-03 is effective for fiscal years beginning after December 15, 2026, and interim periods within fiscal years beginning after December 15, 2027. See BDO's Bulletin, [FASB Finalizes ASU to Disaggregate Income Statement Expenses](#).



Chapter 5: Transition



5.1 OVERVIEW

ASU 2025-06 applies to all entities whether public or private. This chapter includes the related transition guidance and is organized as follows:

- ▶ Effective Date
- ▶ Transition Approaches
- ▶ Transition Disclosures

5.2 EFFECTIVE DATE



FASB REFERENCES

ASC 350-40-65-4(a) and (b)

All entities must apply ASU 2025-06 for annual reporting periods beginning after December 15, 2027, and interim reporting periods within those annual reporting periods.

Early adoption is permitted in an interim or annual reporting period in which financial statements have not yet been issued or made available for issuance. If an entity adopts ASU 2025-06 in an interim reporting period, the adoption must be as of the beginning of the annual reporting period that includes that interim reporting period.

BDO INSIGHTS: NO DELAY IN ADOPTION DATE FOR PRIVATE COMPANIES

ASU 2025-06 is effective for annual reporting periods beginning after December 15, 2027, including interim periods within those annual periods, for **all** entities. The FASB delayed⁴ the effective date to accommodate public business entities' adoption of other ASUs, primarily the new disclosures of disaggregated expense information in ASU 2024-03. Thus, while comment letters generally supported providing additional time for private companies to adopt ASU 2025-06, the FASB concluded that an additional delay was unnecessary as many private companies do not issue U.S. GAAP-compliant financial statements on an interim basis and the changes in ASU 2025-06 are not expected to require significant time or costs to implement.

5.3 TRANSITION APPROACHES

**FASB REFERENCES**

ASC 350-40-65-4(c)

An entity can adopt ASU 2025-06 by using a prospective transition approach, a retrospective transition approach, or a modified transition approach. When selecting a transition approach, an entity may want to consider the status of its software projects and whether software costs were capitalized before the date of adoption to identify an approach that is ideal for its circumstances in terms of time and resources.

5.3.1 Prospective Approach

Under the prospective approach, an entity applies ASU 2025-06 prospectively to new software costs incurred for all projects, **including costs incurred for in-process projects**, during annual reporting periods (and interim reporting periods within those annual reporting periods) beginning after the date of adoption.

BDO INSIGHTS: POTENTIAL CHALLENGES OF PROSPECTIVE APPROACH

While the prospective approach is the most operable transition approach, it could result in:

- ▶ Multiple capitalization models being applied to a single software asset.
- ▶ Impairment charges being recognized for an in-process project for which software costs were capitalized before the date of adoption but the project does not meet the requirements for capitalization under the amendments.

⁴ BC108 through BC110 of ASU 2025-06.

5.3.2 Modified Prospective Approach

Under the modified prospective approach, an entity applies ASU 2025-06 as follows:

For software projects that are either: ▶ New projects ▶ In-process projects, excluding those described below	An entity applies ASU 2025-06 prospectively to all new software costs during annual reporting periods (and interim reporting periods within those annual reporting periods) beginning after the date of adoption.
For in-process software projects that both: ▶ Do not meet the capitalization requirements in Section 3.2 as of the date of adoption AND ▶ Met the capitalization requirements under the previous guidance before the date of adoption	An entity derecognizes capitalized costs for an in-process project through a cumulative-effect adjustment to the opening balance of retained earnings (or other appropriate components of equity or net assets in the statement of financial position) as of the beginning of the annual reporting period in which ASU 2025-06 is adopted.

BDO INSIGHTS: BENEFITS OF MODIFIED PROSPECTIVE APPROACH

The benefits of the modified prospective approach include:

- ▶ It allows entities to align the accounting for ongoing projects without requiring them to identify detailed information about historical software projects that were completed.
- ▶ It addresses the impairment challenges for an in-process project for which capitalization has begun before the adoption date and that project does not meet the requirements for capitalization under the amendments which is an application issue under the prospective approach (see Section 5.3.1).

5.3.3 Retrospective Approach

Under the retrospective transition approach, an entity applies ASU 2025-06 retrospectively through a cumulative-effect adjustment to the opening balance of retained earnings (or other appropriate components of equity or net assets in the statement of financial position) as of the beginning of the first period presented.

BDO INSIGHTS: CHALLENGES OF MODIFIED PROSPECTIVE APPROACH

While the retrospective approach results in comparable financial information across all periods presented, practical challenges in its implementation could arise because:

- ▶ An entity may not have enough detail about historical software costs to adjust prior periods.
- ▶ It could be challenging for an entity to assess and conclude on significant development uncertainty in retrospective periods without applying hindsight.

5.4 TRANSITION DISCLOSURES



FASB REFERENCES

ASC 350-40-65-4(d) through (f)

An entity is required to provide the transition disclosures based on the transition method elected as discussed below.

5.4.1 Prospective Approach

An entity that adopts ASU 2025-06 using the prospective approach is required to disclose the nature of and reason for the change in accounting principle in accordance with ASC 250-10-50-1(a) in both the interim reporting period (if applicable) and the annual reporting period of the change.

5.4.2 Modified Prospective Approach

An entity that adopts ASU 2025-06 using the modified prospective approach is required to disclose **both**:

- ▶ The nature of and reason for the change in accounting principle in accordance with ASC 250-10-50-1(a)
- ▶ The cumulative effect of the change on retained earnings (or other components of equity or net assets in the statement of financial position) as of the beginning of the annual reporting period in which ASU 2025-06 is adopted in both the interim reporting period (if applicable) and the annual reporting period of the change.

5.4.3 Retrospective Approach

An entity that adopts ASU 2025-06 using the retrospective approach is required to provide the transition disclosures required by ASC 250-10-50-1(a) through (b)(1), (b)(2) for any prior periods retrospectively adjusted and (b)(3) and (c)(2) in both the interim reporting period (if applicable) and the annual reporting period of the change.

The disclosures in ASC 250-10-50-1(b)(2) and (c)(1) and ASC 250-10-50-3 related to the current periods, which would require an entity to account for software costs under both the previous guidance and new guidance in the current period, are **not** required because they would have been overly burdensome to prepare.

Appendix A: Other BDO Publications



BDO's Professional Practice – Accounting Group has issued the following Blueprints, which provide detailed accounting guidance, interpretive insights, and practical examples on complex areas of U.S. GAAP:

- ▶ [Statement of Cash Flows Under ASC 230](#)
- ▶ [Segment Reporting Under ASC 280](#)
- ▶ [Equity Method Investments Under ASC 323](#)
- ▶ [Revenue Recognition Under ASC 606](#)
- ▶ [Share-based Payments Under ASC 718](#)
- ▶ [Business Combinations Under ASC 805](#)
- ▶ [Control and Consolidation Under ASC 810](#)
- ▶ [Accounting for Leases Under ASC 842](#)
- ▶ [Issuer's Accounting for Complex Financial Instruments](#)

BDO's Professional Practice – SEC Services Group has issued the following Blueprints to help SEC registrants apply the SEC's rules and regulations related to financial reporting:

- ▶ [Guide to Going Public](#)

Series Overview: Building Knowledge

In addition to Blueprints, BDO's Professional Practice has issued a suite of publications related to U.S. GAAP and SEC rules and regulations:

- ▶ [Bulletins](#): short, timely communications delivering essential news, alerts, and developments in accounting and SEC reporting
- ▶ [Snapshots](#): summaries of SEC rules and regulations
- ▶ [SEC Reporting Insights](#), updated annually
- ▶ [Conference Highlights: Current SEC & PCAOB Developments](#), updated annually
- ▶ [Effective Date Reminders](#), updated annually
- ▶ [Industry Supplements](#)
- ▶ [Other publications](#) on new accounting standards and SEC rules and regulations.

For more on IFRS, visit BDO's [IFRS and Corporate Reporting](#) for more information.



Contacts

PROFESSIONAL PRACTICE GROUP CONTACTS:

ANGELA NEWELL
Deputy Chief Accountant
214-689-5669 / ajnewell@bdo.com

ADAM BROWN
Chief Accountant
214-665-0673 / abrown@bdo.com

INDUSTRY GROUP CONTACTS:

HANK GALLIGAN
National Technology Industry Leader
617-422-7521 / hgalligan@bdo.com

STEPHANIE HEWLETT
National Technology Industry Assurance Leader
801-456-5718 / shewlett@bdo.com

FASB publications excerpted in this publication were reprinted with permission. Copyright 2026 by Financial Accounting Foundation, Norwalk, Connecticut.

Our purpose is helping people thrive, every day. Together, we are focused on delivering exceptional and sustainable outcomes and value for our people, our clients and our communities. BDO is proud to be an ESOP company, reflecting a culture that puts people first. BDO professionals provide assurance, tax and advisory services for a diverse range of clients across the U.S. and in over 160 countries through our global organization.

BDO is the brand name for the BDO network and for each of the BDO Member Firms. BDO USA, P.C., a Virginia professional corporation, is the U.S. member of BDO International Limited, a UK company limited by guarantee, and forms part of the international BDO network of independent member firms. For more information, please visit: www.bdo.com.