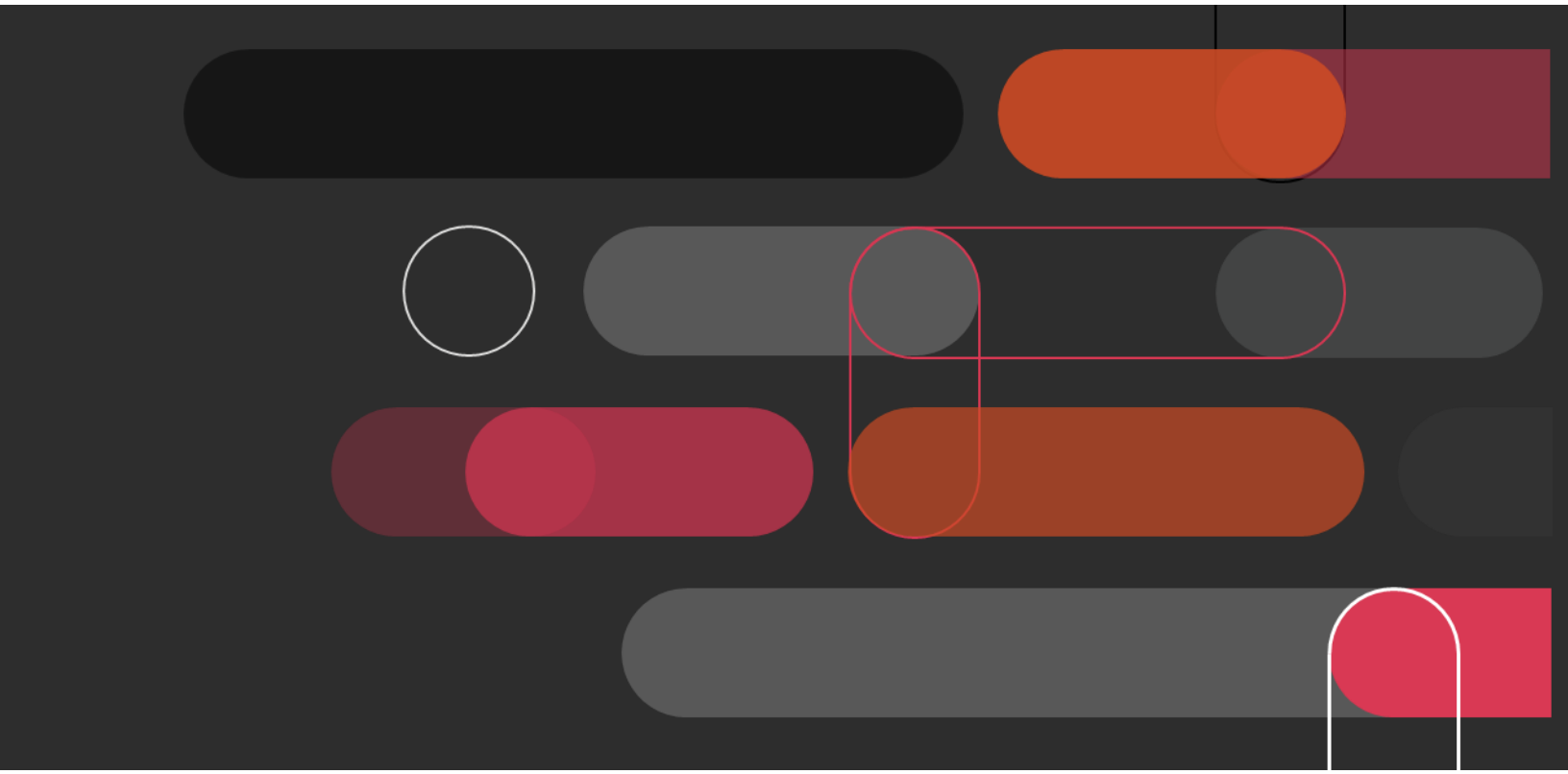
www.viewpoint.pwc.com



National Office

Software costs

Partially updated April 2025

pwc

About the Software costs guide

PwC is pleased to offer the first edition of our *Software costs* guide. This guide discusses the framework or scope for accounting for software and software-related costs, as well as the initial and subsequent accounting for those costs.

This guide summarizes the applicable accounting literature, including relevant references to and excerpts from the FASB's Accounting Standards Codification (the Codification). It also provides our insights and perspectives, interpretative and application guidance, illustrative examples, and discussion on emerging practice issues.

This guide should be used in combination with a thorough analysis of the relevant facts and circumstances, review of the authoritative accounting literature, and appropriate professional and technical advice.

References to US GAAP

Definitions, full paragraphs, and excerpts from the FASB's Accounting Standards Codification are clearly labelled. In some instances, guidance was cited with minor editorial modification to flow in the context of the PwC Guide. The remaining text is PwC's original content.

References to other PwC guidance

This guide provides general and specific references to chapters in other PwC guides to assist users in finding other relevant information. References to other guides are indicated by the applicable guide abbreviation followed by the specific section number. The other PwC guides referred to in this guide, including their abbreviations, are:

- *Business combinations and noncontrolling interests (BCG)*
- *Financial statement presentation (FSP)*
- *Property, plant, equipment and other assets (PPE)*
- *Revenue from contracts with customers (RR)*

Summary of significant changes

Following is a summary of the noteworthy revisions to the guide since it was last updated. Additional updates may be made to keep pace with significant developments.

Revisions made in April 2025

SW 3, *Internal-use software*

- SW 3.3.1.1 was added to discuss costs of developing artificial intelligence (AI) software applications.

Copyrights

This publication has been prepared for general informational purposes, and does not constitute professional advice on facts and circumstances specific to any person or entity. You should not act upon the information contained in this publication without obtaining specific professional advice. No representation or warranty (express or implied) is given as to the accuracy or completeness of the information contained in this publication. The information contained in this publication was not intended or written to be used, and cannot be used, for purposes of avoiding penalties or sanctions imposed by any government or other regulatory body. PricewaterhouseCoopers LLP, its members, employees, and agents shall not be responsible for any loss sustained by any person or entity that relies on the information contained in this publication. Certain aspects of this publication may be superseded as new guidance or interpretations emerge. Financial statement preparers and other users of this publication are therefore cautioned to stay abreast of and carefully evaluate subsequent authoritative and interpretative guidance.

The *FASB Accounting Standards Codification*® material is copyrighted by the Financial Accounting Foundation, 401 Merritt 7, Norwalk, CT 06856, and is reproduced with permission.

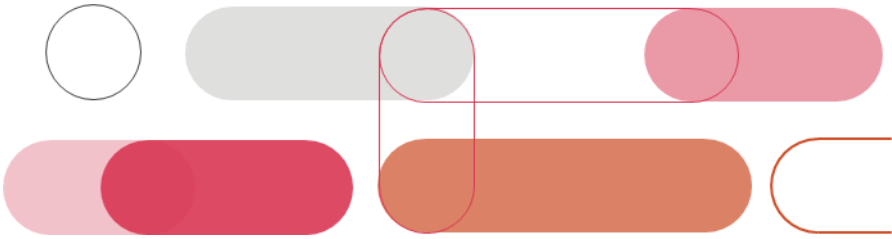
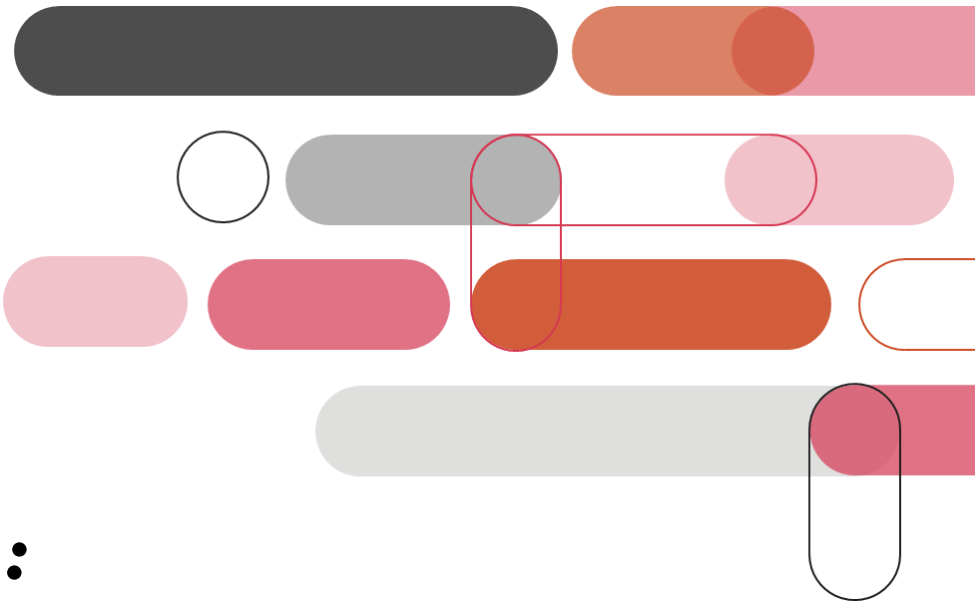


Table of contents

Chapter 4: Cloud computing implementation costs—updated December 2021

Chapter 5: Other software-related costs—updated December 2021



Chapter 1: Framework for accounting for software and software-related costs—updated December 2021

1.1 Software and software-related costs—overview—updated April 2025

The accounting for software and software-related costs is largely modeled after inventory guidance (for software that will be sold to customers) or fixed assets guidance (for software that will be used internally). The internal-use software guidance also addresses situations when a reporting entity is accessing software through a hosting arrangement (e.g., purchasing software as a service (SaaS)) and provides criteria to determine whether a reporting entity is obtaining software (an asset) or solely receiving a service. Software development costs span the spectrum from research & development costs, which are expensed as incurred, through the costs to create a self-constructed asset to the costs of purchasing an intangible asset.

Artificial intelligence (AI) refers to programs' and machines' ability to complete tasks that normally require human intelligence. A component of AI includes a class of software that use coded algorithms interacting with datasets to problem solve. While AI software programs have unique capabilities, an accounting standard that provides specific guidance related to the accounting for the costs of developing AI software programs does not exist. Thus, the existing software guidance, which is described below and discussed in more detail throughout this guide, applies equally to the costs of developing AI software. While there will likely be many similarities in the types of costs incurred when developing AI software as compared to other software products, AI software development may involve types of costs and activities that are unique to, or more prevalent in, AI software program development.

ASC 985-20, *Software—Costs of Software to Be Sold, Leased, or Marketed,* applies to development costs incurred for software to be sold, leased, or otherwise marketed as a separate product or embedded within a product or process. It addresses which software costs are to be capitalized and how the software asset should be derecognized and recognized as cost of revenue or cost of sales. The revenue generated from licensing software is subject to the revenue recognition guidance in ASC 606, *Revenue from Contracts with Customers*. For guidance on accounting for revenue from software to be sold, leased, or otherwise marketed, refer to PwC's *Revenue from contracts with customers* guide.

ASC 350-40, *Intangibles—Goodwill and Other—Internal-Use Software,* applies to software that is acquired, internally developed, or modified solely to meet the reporting entity's internal needs. Software used to provide a service to a customer (e.g., through a SaaS arrangement) is software that is considered to meet an internal need of the service provider and should be accounted for as internal-use software. ASC 350-40 also includes guidance on a customer's accounting for implementation, setup, and other upfront costs (collectively implementation costs) incurred when entering into a cloud computing arrangement that is a service contract.

Note about ongoing standard setting

The FASB has an active project related to the accounting and disclosure of software costs. Financial statement preparers and other users of this publication are therefore encouraged to monitor the status of the project, and if finalized, evaluate the effective date of the new guidance and the implications on the accounting and related disclosures.

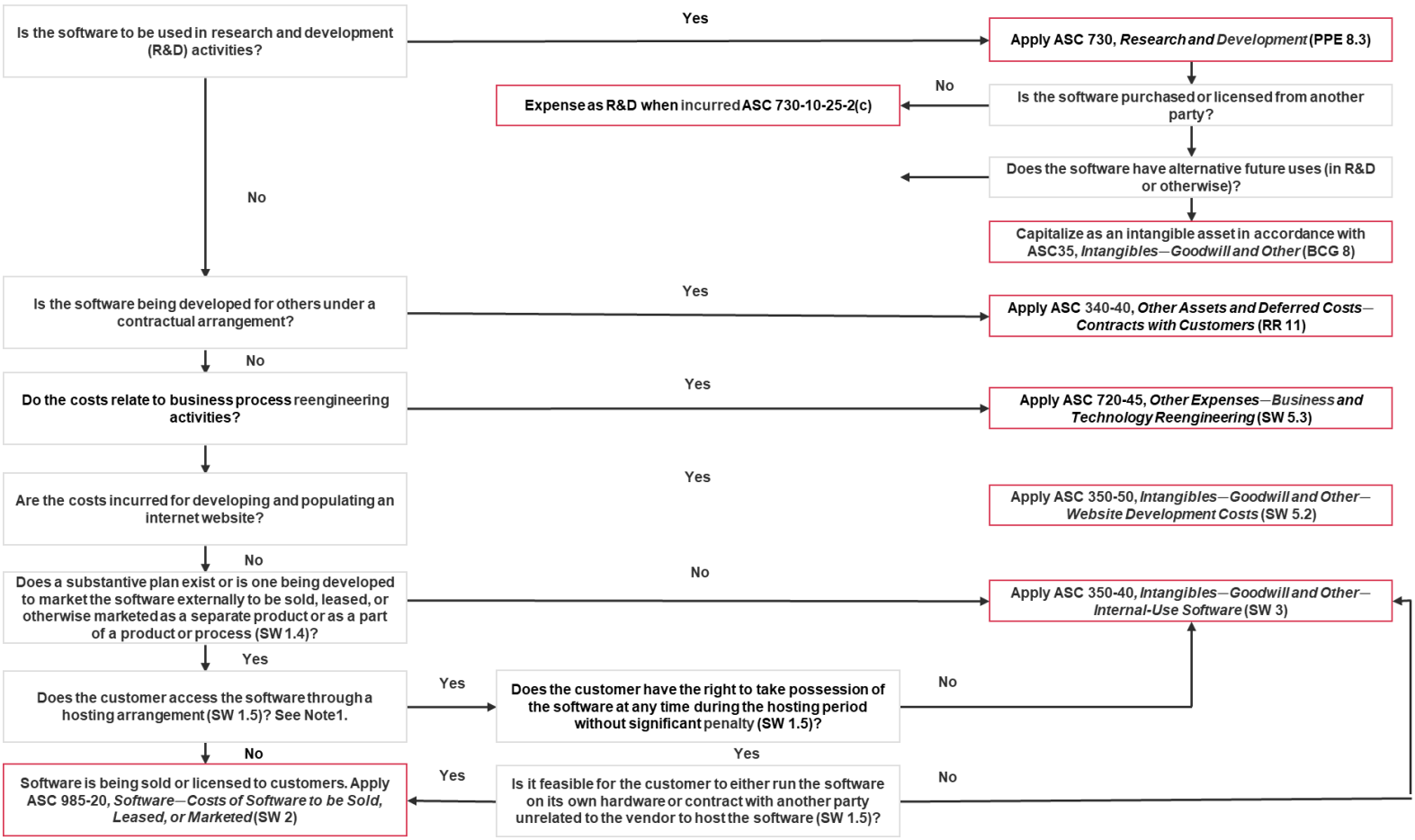
1.2 Scope of software cost guidance

Specific guidance exists to address the accounting for costs related to software that is (1) sold, leased, or otherwise marketed (see SW 2), (2) developed or obtained for

internal use (see SW 3), and (3) accessed through the cloud (see SW 4). Other types of software-related costs that could qualify for capitalization, such as website development costs and costs of business process reengineering activities are discussed in SW 5.

The decision tree in Figure SW 1-1 walks through the key factors that determine the relevant guidance for costs incurred by vendors to obtain or develop software.

FIGURE SW 1-1
Determining the relevant guidance for costs incurred by vendors to obtain or develop software



Note 1- Refer to SW 1.5.2 for situations when the vendor both (a) licenses the software and (b) provides access to it through a hosting arrangement.

1.3 *Externally marketed software–scope*

ASC 985-20 establishes the accounting and reporting for the costs of 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 or purchased. ASC 985-20 applies to development costs for software that will be separately licensed to customers (e.g., on-premises software), as well as software included within a product sold to a customer. For example, ASC 985-20 applies to software embedded in a semiconductor chip that is used as a component of a product that is

sold to a customer (e.g., automobile electronic systems). Software that is embedded in a product is sometimes referred to as “firmware.”

Typically, software that is used by the vendor in the production of a good or for providing a service is not considered externally marketed software unless that software is included in or part of the actual good or service sold. For example, ASC 985-20 does not apply to software that is solely used to provide SaaS or similar services to a customer when: (a) the customer does not have the contractual right to take possession of the software or (b) it is not feasible for the customer to run the software on its own (see SW 1.5). However, if software is transferred to the customer that is used in conjunction with services provided by the reporting entity (e.g., the customer uses on-premises software to collect data that is monitored by the reporting entity), the related software development costs would be in the scope of ASC 985-20.

Software that is not intended to be sold, leased, or otherwise marketed separately or as part of a product or process should be accounted for in accordance with the guidance for internal-use software in ASC 350-40. Refer to SW 1.4 for further discussion of the distinction between externally marketed software and internal-use software.

1.4 Internal-use software—scope

ASC 350-40 provides guidance on accounting for the costs of software developed or obtained for internal use (internal-use software).

ASC 350-40-15-2A

Internal-use software has both of the following characteristics:

a. The software is acquired, internally developed, or modified solely to meet the entity’s internal needs.

b. During the software’s development or modification, no substantive plan exists or is being developed to market the software externally.

A plan to market software externally would be considered substantive if its implementation is reasonably possible. A substantive plan could include the selection of marketing channels with identified promotional, delivery, billing, and support activities. Typically, reporting entities planning to sell or license software externally would have potential customers identified and resources committed to a salesforce and marketing activities.

In some circumstances, a reporting entity that previously had no plan to license internal-use software to other parties will subsequently decide to license or sell that software. In that situation, the guidance in ASC 985-20 for externally marketed software should be followed on a prospective basis. Further, if a reporting entity sells software to a customer after the development of the internal-use software is completed, proceeds from that sale should be offset against the carrying amount of the internal-use software asset (ASC 350-40-35-7). Refer to SW 3.9 for further guidance on the accounting implications when a reporting entity decides to market internal-use software to customers.

If a reporting entity has a pattern of selling software to a third party that was originally being developed to use internally, there is a rebuttable presumption that any software developed by that reporting entity is intended for sale, lease, or other marketing (ASC 350-40-15-2C).

ASC 350-40-55-1 provides examples of scenarios when software is considered to be developed for internal use. ASC 350-40-55-2 provides examples of scenarios when software is considered to be developed for something other than internal use. While the lists of examples are lengthy, they are neither all-inclusive or determinative. Judgment will be required to make the determination in some cases.

The primary consideration for determining whether software is developed or obtained for internal use is whether the software will ultimately be transferred to a customer (either on its own or integrated as part of a product or process). For example, a reporting entity providing SaaS to a customer when the customer cannot take possession of the software or run the software without the vendor's infrastructure (see SW 1.5) does not transfer the software to the customer and is only providing a service. Thus, the reporting entity would apply the guidance for software developed for internal use in ASC 350-40 to the related software costs.

If software is transferred to a customer, it is generally not considered to have been developed solely to meet the reporting entity's internal needs (and, therefore, the related costs are subject to the guidance on externally marketed software in ASC 985-20). However, if the only purpose of the software is to enable connection to a cloud-based service from the reporting entity, it may be appropriate to conclude the software is not "externally marketed" and therefore, apply ASC 350-40 to the related development costs. This conclusion would not be appropriate if the software transferred to the customer has substantive standalone functionality.

QUESTION SW 1-1

Which guidance applies to development costs related to mobile applications (a form of software) that are downloaded by customers?

PwC response

It depends. Mobile applications that are sold to customers, such as a mobile game application (whether downloaded for a fee or monetized through in-app purchases) are generally considered externally marketed software; therefore, the related development costs would be in the scope of ASC 985-20. In contrast, mobile applications that are provided to customers only to enable connection to the reporting entity's cloud-based services, such as a mobile banking application, would generally not be considered externally marketed software. Therefore, the related development costs would be in the scope of ASC 350-40. Determining the appropriate scope for costs related to mobile applications could require judgment.

1.5 Cloud computing arrangements—scope

As companies move their data, applications, and platforms to the cloud, software that a customer would have traditionally installed locally on its own servers is now often hosted on the vendor's (or a third-party cloud platform provider's) servers and accessed by the customer remotely. These arrangements are sometimes referred to as hosting arrangements or cloud computing arrangements (CCAs). Examples of CCAs include SaaS and other "as a service" arrangements, including platform as a service and infrastructure as a service. Some CCAs transfer a license to the software in addition to the service of hosting the software. It is important to determine whether, for accounting purposes, the CCA includes a software license in addition to the service or if it is only a service. ASC 350-40-15-4A contains the guidance for making this assessment for both vendors and customers in a CCA.

ASC 350-40-15-4A

The guidance in the General Subsections of this Subtopic applies only to internal-use software that a customer obtains access to in a hosting arrangement if both of the following criteria are met:

a. The customer has the contractual right to take possession of the software at any time during the hosting period without significant penalty.

b. 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.

The right to take possession of the software at any time during the hosting period can be explicit, or implicit if the vendor creates a valid expectation that it will provide the customer with the right.

Determining whether the customer has the right to take possession of the software without significant penalty can require judgment. The phrase "without significant penalty" refers to the ability to (1) take delivery of the software without incurring significant cost and (2) use the software separately without a significant diminution in utility or value. Significant costs could include penalties that the customer would incur for terminating the hosting arrangement (including forfeiting a nonrefundable upfront payment for services), as well as costs the customer would incur for the necessary infrastructure to host the software itself. "Significant diminution in utility or value" generally refers to a reduction in features, functions, processing speed, or computing power of the software, including losing the right to receive upgrades or updates that are integral to the functionality of the software.

If both of the criteria in ASC 350-40-15-4A are met, the CCA includes a software license to be accounted for in addition to a hosting service. A CCA that does not meet those criteria should be accounted for solely as a service contract, regardless of whether the contract includes language that refers to a software license.

1.5.1 *Cloud computing arrangements—customer's accounting*

A reporting entity that is the customer (i.e., a purchaser) of a CCA provided by a third-party vendor may be entering into an arrangement that is solely a service or acquiring a software license in addition to a service.

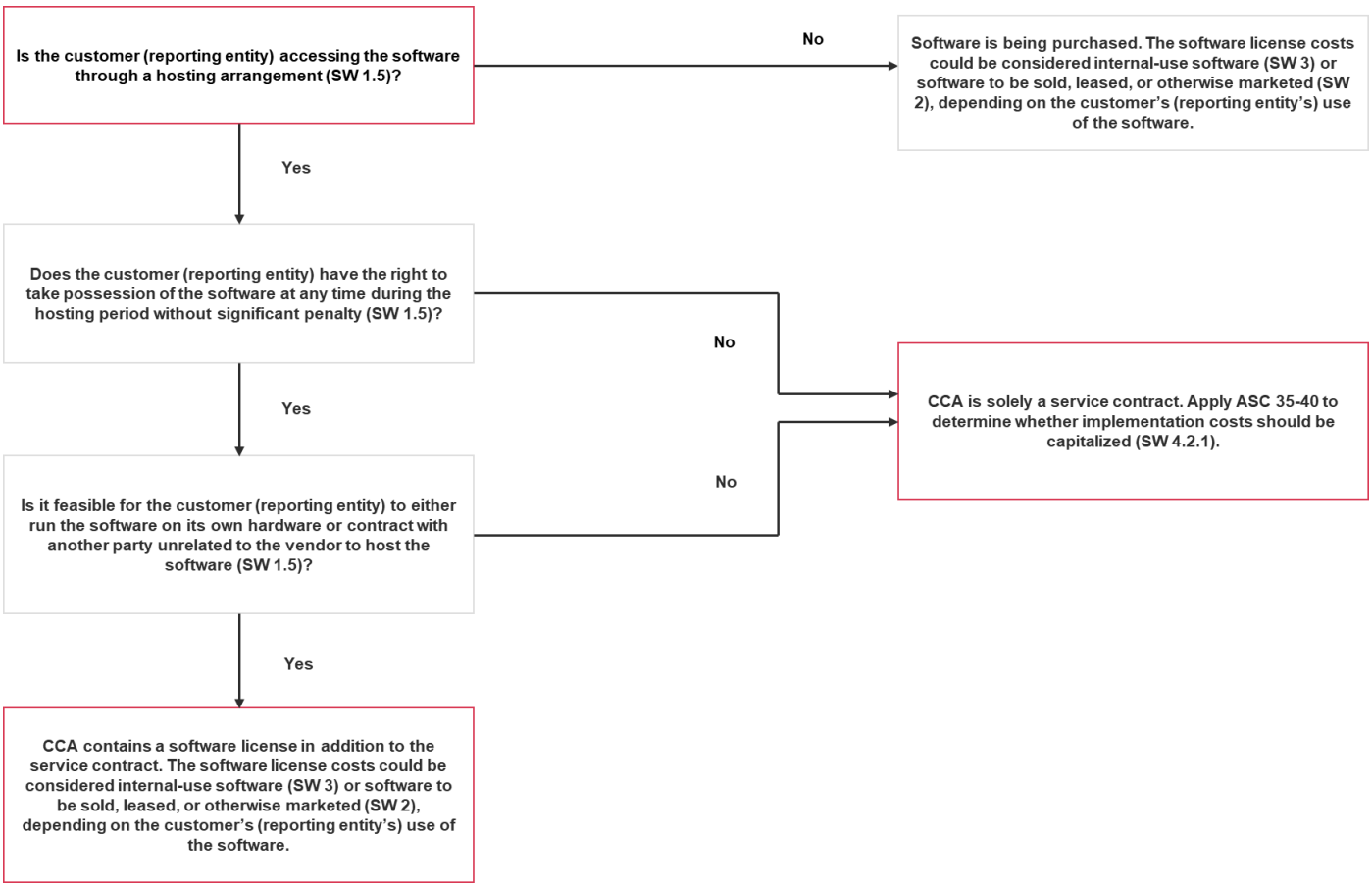
- **CCA is solely a service**: Although the reporting entity is not acquiring software in this situation, the determination of whether to capitalize the related implementation costs is based on the internal-use software guidance, as discussed in SW 4. The ongoing cost for CCA services is expensed as the service is received similar to other service contracts.

- **CCA includes a software license**: The costs to acquire the software license could be in the scope of the internal-use software or externally marketed software guidance, depending on the reporting entity's use of the software. Refer to SW 1.3 and SW 1.4 for scoping considerations for software costs.

It is common for arrangements with third-party vendors to include multiple elements, such as a software license, SaaS, and other services. Refer to SW 3.6 for guidance on allocating costs related to a multiple-element contract with a third-party vendor.

Figure SW 1-2 details the relevant considerations for determining the appropriate guidance for costs incurred by a reporting entity that is the customer for a CCA.

FIGURE SW 1-2
Determining the relevant guidance for costs incurred by a reporting entity that is the customer in a CCA



1.5.2 *Cloud computing arrangements—vendor's accounting*

A vendor providing a CCA to customers will need to assess whether the arrangement includes a license (i.e., whether the software is transferred to the customer) to determine which guidance to apply to the related software development costs. The guidance in ASC 350-40-15-4A (see SW 1.5) is used to make this assessment for both vendors and customers in a CCA.

- **CCA is solely a service:** As the software is solely used by the vendor to provide a service to customers, the related software development costs should be accounted for under the internal-use software guidance (see SW 3).
- **CCA includes a software license:** The software development costs incurred by the vendor for software being licensed to customers should be accounted for under the externally marketed software guidance (see SW 2).

Refer to Figure SW 1-1 for a decision tree for determining the relevant guidance for costs incurred by vendors to obtain or develop software, including software a customer accesses through a hosting arrangement.

QUESTION SW 1-2

Which guidance applies to software development costs related to software that a reporting entity uses to provide a service to some customers in a hosting arrangement, but licenses it to other customers (as on-premises software)?

PwC response

Generally, if a reporting entity has substantive sales of on-premises software, the guidance for externally marketed software in ASC 985-20 should be applied to the related software development costs. This is the case even if the reporting entity is also using the same software to provide a service to customers in a hosting arrangement. If the reporting entity incurs costs that relate solely to the platform used for hosting the software or only the hosted version of the software (and not the licensed on-premises software), the guidance for internal-use software in ASC 350-40 may apply to those costs. Assessing which costs, if any, are subject to the internal-use software guidance in this situation may require judgment.

QUESTION SW 1-3

What are the accounting implications if a reporting entity has capitalized costs under ASC 350-40 related to software it uses to provide a service (as a SaaS arrangement), but subsequently agrees to license the same software to a customer that wants to host the software itself?

PwC response

If an entity decides to sell software that was capitalized under ASC 350-40 based on an assumption that the software would be used only for internal use, the proceeds from the sale are applied against the carrying amount of that software in accordance with the guidance in ASC 350-40-35-7. Refer to SW 3.9 for further discussion of the accounting implications in these circumstances. The reporting entity should also consider whether future development costs (e.g., product enhancements or new software products) should be accounted for under ASC 985-20. As discussed in SW 1.4, ASC 350-40-15-2C indicates that a pattern of selling software to a third party that was originally developed to use internally creates a rebuttable presumption that any software developed by that reporting entity is intended for sale, lease, or other external marketing.

QUESTION SW 1-4

What are the accounting implications if a reporting entity has capitalized costs under ASC 985-20 related to software that it both (a) licenses to customers (as on-premises software) and (b) uses to provide a service (a SaaS arrangement), but subsequently discontinues licensing the software to customers?

PwC response

Many reporting entities are transitioning from licensing on-premises software to providing SaaS offerings to customers. The accounting guidance does not specifically address the transition from externally marketed software to internal-use software. However, a reporting entity may conclude that it no longer intends to market software externally even though it has a history of licensing software to customers. As a result, development costs for new software products or enhancements to existing products (unless licensed for on-premises use by customers) may be in the scope of ASC 350-40. This determination could require significant judgment. If the reporting entity is transitioning to SaaS offerings, but expects to continue to have substantive licenses of on-premises software, ASC 985-20 should continue to be applied.

QUESTION SW 1-5

Which guidance applies to development costs related to software that will be hosted by a third-party service provider (i.e., a party other than the reporting entity or customer) in arrangements with customers?

PwC response

It depends on whether the third party is hosting the software on behalf of the reporting entity (the vendor) or the customer. If the reporting entity contracts with a third party to host software developed by the reporting entity (to be accessed by customers), the accounting for the related software development costs is no different than the reporting entity hosting the software on its own servers. The reporting entity should apply the guidance in ASC 350-40-15-4A (see SW 1.5) to assess whether the arrangement with the customer includes a license, which will determine which software cost guidance applies (see Figure SW 1-1).

If the customer purchasing the software chooses to utilize a third-party hosting service instead of hosting the software on its own servers, the reporting entity could conclude it has transferred the software to the customer. Thus, the related development costs would be accounted for under ASC 985-20.

QUESTION SW 1-6

Which guidance applies to costs incurred by a reporting entity (a vendor) to implement a software product or a CCA for a specific customer?

PwC response

Costs incurred to implement a software product or CCA for a specific customer are generally costs to fulfill a revenue contract and would be subject to the guidance in ASC 340-40, *Other Assets and Deferred Costs—Contracts with Customers.* In contrast, if the reporting entity incurs costs to make modifications to the software that benefit multiple customers, the reporting entity should consider the guidance in internal-use software or externally marketed software, as applicable.

1.5.2.1 *Hybrid cloud arrangements*

An arrangement that includes both a license to on-premises software and cloud-based services (such as SaaS) is sometimes referred to as a "hybrid" or "hybrid cloud" arrangement. While the term "hybrid cloud" may be used to refer to various types of arrangements that involve the cloud, this discussion refers specifically to arrangements that include both an on-premises software component and a cloud-based services component.

Typically, in a hybrid cloud arrangement, the on-premises software has standalone functionality and incremental functionality is obtained by accessing cloud-based services. For purposes of revenue recognition, a reporting entity that provides hybrid cloud offerings to customers may conclude, depending on the facts and circumstances, that (a) the license to on-premises software and the cloud-based services are each distinct or (b) the license to on-premises software and the cloud-based services comprise a single performance obligation. Refer to RR 9.3 for guidance on this assessment.

Generally, the development costs related to the licensed on-premises software transferred to customers in a hybrid cloud arrangement are subject to the guidance for externally marketed software in ASC 985-20. Costs that relate solely to the software used to provide cloud-based services would generally be subject to the guidance for internal-use software in ASC 350-40, unless the same software product

is also licensed to customers, in which case the guidance for externally marketed software in ASC 985-20 would apply. Judgment may be required to distinguish between costs that relate to the licensed software and costs related solely to software used to provide a service.

The assessment of which guidance applies to software development costs related to a hybrid cloud arrangement is generally not impacted by the reporting entity's conclusions regarding whether the software license is distinct for revenue recognition purposes. For example, the guidance for externally marketed software in ASC 985-20 would likely apply to a software license transferred to a customer even if it is not distinct as a result of being highly interdependent or highly interrelated with cloud-based services. As discussed in SW 1.3, ASC 985-20 applies to software that is sold to customers either as separate product or "as part of a product or process;" accordingly, the guidance is not limited to arrangements in which the reporting entity concludes the software is distinct. An exception would be an arrangement in which the software's only purpose is to enable connection to the cloud-based service (see SW 1.4); however, this is not typically the case in hybrid cloud offerings.

1.6 Software acquired in a business combination

It is common for software to be an identified intangible asset of the acquired entity in a business combination. ASC 805, *Business Combinations*, provides guidance on the identification and initial measurement of the identified intangible assets acquired in a business combination (refer to BCG 4); however, ASC 805-10-35-1 specifically indicates that the subsequent accounting is based on other GAAP, depending on the nature of the acquired asset.

Excerpt from ASC 805-10-35-1

In general, an acquirer shall subsequently measure and account for assets acquired, liabilities assumed or incurred, and equity instruments issued in a business combination in accordance with other applicable generally accepted accounting principles (GAAP) for those items, depending on their nature.

Once the fair value of any software acquired is measured and recorded upon acquisition, the guidance in ASC 985-20 or ASC 350-40 will apply to the post-acquisition accounting for the software depending on how the reporting entity uses the software, as discussed in SW 1.3 and SW 1.4. This determination is important because the post-acquisition accounting (including amortization and impairment), presentation, and disclosure requirements differ between software that is externally marketed and internal-use software.

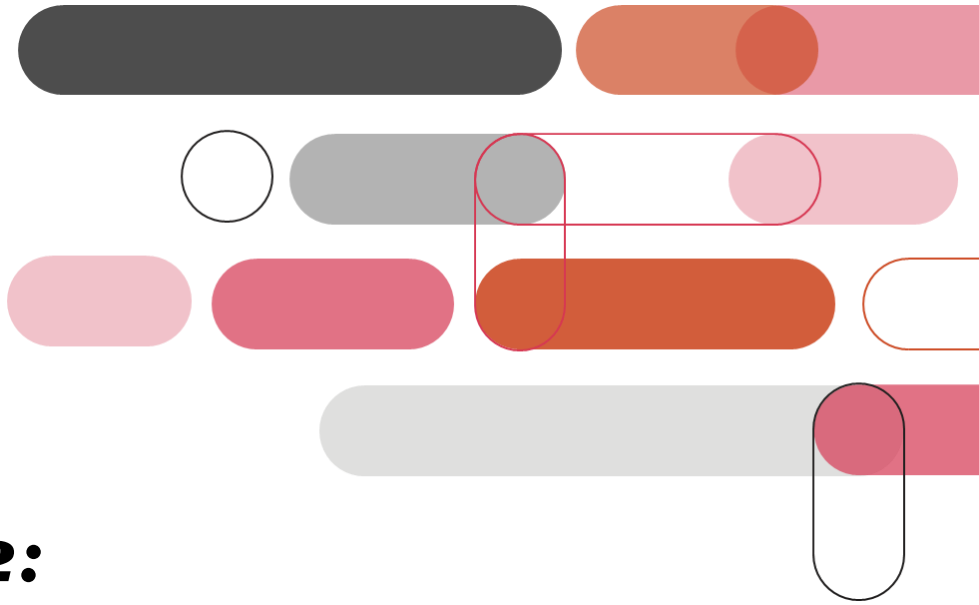
1.7 Software costs—presentation and disclosure

The presentation and disclosure of software-related costs in the financial statements depends on the nature of the costs and how the software will be used. As described in SW 1.1, the accounting guidance for software-related costs is generally modeled after the inventory guidance (for software that is sold to customers) or the fixed assets guidance (for software that is used internally); accordingly, the presentation of those respective costs will generally be similar to those comparable financial statements line items.

For a CCA that is a service contract, although the determination of whether costs should be capitalized is based on the internal-use software guidance, the presentation of costs is similar to other purchased services. For example, the

amortization of capitalized implementation costs is presented in the same line item as the ongoing costs of the CCA service.

See FSP 8.6 for further discussion of presentation and disclosure requirements for capitalized software and software-related costs.



Chapter 2: Software to be sold, leased, or marketed—updated December 2021

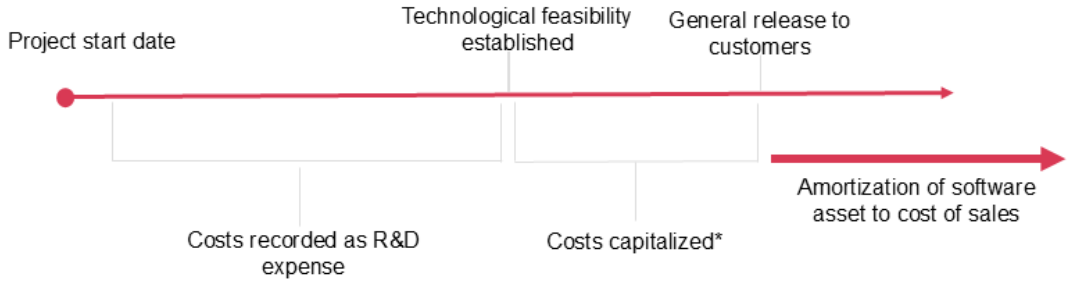
2.1 *Software to be sold, leased, or marketed—chapter overview*

ASC 985-20 applies to costs to internally develop and produce, or to purchase, software that the vendor intends to sell, lease, or otherwise market externally—either separately or as part of a product (externally marketed software). See SW 1.3 for guidance on assessing whether software is considered externally marketed and thus, in the scope of ASC 985-20.

The guidance in ASC 985-20 delineates the software development costs to be expensed from those to be capitalized based on whether the costs are incurred before or after the product reaches technological feasibility. Costs incurred prior to reaching technological feasibility are considered research and development (R&D) costs and expensed when incurred as required by ASC 730-10, *Research and Development—Overall*. Costs to develop, produce, or purchase externally marketed software are capitalized once technological feasibility is reached. Capitalization of such costs ceases when the product is available for general release. Costs of subsequent product enhancements are also included in the scope of ASC 985-20 and may qualify for cost capitalization (see SW 2.9). The types of costs that should be capitalized are similar to those that would be capitalized during the production of inventory as described in ASC 330, *Inventory* (see SW 2.3).

Figure SW 2-1 illustrates an example project timeline for the development of externally marketed software.

FIGURE SW 2-1
Externally marketed software – example development timeline



** Certain costs are expensed as incurred. See SW 2.3 for discussion of capitalizable costs.*

2.2 *Establishing technological feasibility*

ASC 985-20-25-1 requires a reporting entity to expense when incurred all costs prior to establishing technological feasibility of externally marketed software. Those costs are considered R&D expense. The determination of what constitutes technological feasibility, and the point at which it is established are, therefore, critical determinations in the accounting for the costs of externally marketed software. The criteria for establishing technological feasibility are discussed in ASC 985-20-25-2.

ASC 985-20-25-2

For purposes of this Subtopic, 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. At a minimum, the entity shall have performed the activities in either (a) or (b) as evidence that technological feasibility has been established:

a. If the process of creating the computer software product includes a detail program design, all of the following:

   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.

   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.

   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.

b. If the process of creating the computer software product does not include a detail program design with the features identified in (a), both of the following:

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

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

As ASC 985-20-25-2 indicates, technological feasibility is established through completion of either (a) a detailed program design or (b) a working model. This is not an accounting policy election. Rather, which milestone establishes technological feasibility is a function of the reporting entity's software development process for each project. In practice, many software companies define technological feasibility as the creation of a working model, which often occurs late in the development cycle.

ASC 985-20-55-7 addresses how to determine technological feasibility when a software product comprises various modules that are not sold separately.

ASC 985-20-55-7

When a product comprises various modules that are not separately saleable, technological feasibility is established for the product as a whole, 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.

2.2.1 *Technological feasibility—detailed program design*

As detailed in ASC 985-20-25-2, in order to achieve technological feasibility by reference to a detailed program design, (a) the product design must be complete, (b) all resources necessary to produce the product must be available, (c) the completeness of the detailed program design must have been confirmed by documenting and tracing to product specifications, and (d) all high-risk development issues must have been resolved through coding and testing.

In the unusual case when a high-risk development issue arises after management has established technological feasibility, the provisions of ASC 250-10-45-17 for changes in accounting estimates would apply to the previously capitalized costs, as well as the costs to resolve any high-risk development issues. Any previously

capitalized costs for that product, as well as any additional costs incurred to re-establish technological feasibility, should be charged to expense as R&D costs.

Detailed program design requirements are generally not met simply by a chart outlining work plan tasks or an overview flowchart or diagram of the product. Conceptually, a detailed program design is analogous to an engineering blueprint.

In each development environment, the form of a detailed program design will vary; however, it should typically include the following:

- A detailed description of product specifications (the product design)
- Detailed program design and flowcharts documenting the program procedures, data flow, and interaction with other applications. This documentation should take product function, feature, and technical requirements to their most detailed, logical form and should be ready for coding.
- Documentation of the consideration and resolution of high-risk development issues through coding and testing
- Actual coding and testing of specific program sections, when warranted

Once technological feasibility has been attained by development of an appropriately detailed program design, capitalization of software costs should begin. In other words, reporting entities cannot elect to delay capitalization until a working model has been completed.

2.2.2 *Technological feasibility—working model*

If the development process does not include a detail program design with the characteristics described in SW 2.2.1, technological feasibility is established by a working model. ASC 985-20 requires that the completeness of the working model and its consistency with the product design be confirmed by testing before capitalization begins. A working model is not the same as a prototype; a working model is typically only available near the end of the development process and is generally the beta testing version of the software.

Definition from ASC Master Glossary

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).

2.3 *Capitalizable costs for externally marketed software—updated April 2025*

Once technological feasibility is established, software development costs that directly relate to the project should be capitalized until the product is available for general release to customers, which is when the software or the product that contains the software is available for purchase. ASC 985-20-25-2 through ASC 985-20-25-5 discusses the types of costs that should be capitalized once technological feasibility is established, including costs related to coding and testing activities. While ASC 985-20 does not include examples of direct costs, we believe the examples provided in ASC 350-40-30-1 are also relevant to externally marketed software (see SW 3.3). These include payroll and payroll-related costs for employee's time spent on an

internal-use software project, travel expenses incurred by employees directly associated with developing software, and fees paid to third parties for development services. See also SW 3.3.1.1 for discussion of costs prevalent in AI software development, including development of the algorithm and purchased data. When such costs are incurred in the development of externally marketed software, they should be considered in applying the guidance in ASC 958-20-25.

Indirect costs also qualify for capitalization, such as overhead related to programmers and the dedicated computer hardware and systems they utilize or an allocation of the cost of intangible assets acquired to train a large language model. However, an allocation of general and administrative expenses unrelated to the individuals or the activities associated with the software development project would not be appropriate. In contrast to ASC 985-20, ASC 350-40-30-2 does not permit capitalization of any overhead costs (see SW 3.3.1).

QUESTION SW 2-1

May a reporting entity elect to expense all software development costs for externally marketed software?

PwC response

No. A reporting entity should capitalize those costs that meet the criteria in ASC 985-20 for capitalization. The capitalization of software costs is not optional. However, in practice, many reporting entities' business practices result in technological feasibility being established based on a working model, which is typically only available near the end of the development process. As a result, for some reporting entities, costs incurred after technological feasibility is established but before general release to customers may be nominal.

2.4 Agile development for externally marketed software

One of the key challenges in accounting for software development costs stems from the continuous technological changes in software development practices, which now largely follow agile development principles. An agile development process uses sprints for planning and execution in which a larger project is typically broken down into smaller increments and feedback is used to continuously iterate the development process, resulting in features and functionalities that are individually developed and continuously changing. This is in contrast to a linear development process, which typically involves the development of a detailed program design, a development plan, and a sequential approach to coding.

Reporting entities using an agile development process are nevertheless subject to the guidance in ASC 985-20. Specifically, capitalization of costs commences once technological feasibility is established. In practice, reporting entities that employ an agile development process typically do not complete a detailed program design sufficient to establish technological feasibility. Thus, these reporting entities will often utilize a working model to establish technological feasibility, which generally results in costs being capitalized later in the development process.

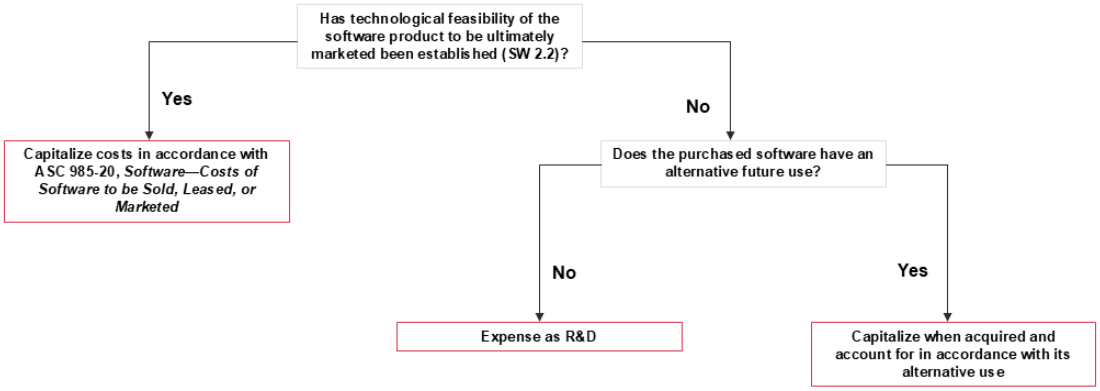
2.5 Purchased software to be externally marketed

The cost of purchased software is generally accounted for in the same manner as costs incurred to develop such software internally unless it is acquired in a business combination. If technological feasibility of the software product to be ultimately marketed has been established, the cost of purchased software should be

capitalized. If technological feasibility has not been established, the costs are expensed as R&D, unless the purchased software has an alternative use. See BCG 4 for information on in-process R&D assets acquired in a business combination.

Figure SW 2-2 provides a summary of accounting for purchased software.

FIGURE SW 2-2
Accounting for purchased software



The cost to purchase a completed software product that will undergo minimal changes before it is marketed to customers will generally be capitalized in its entirety. In that case, the software has already reached technological feasibility and the purchase cost is considered a cost incurred after technological feasibility is established. If the purchased software will become part of the ultimate software product to be marketed to customers, reporting entities may need to apply judgment to determine whether the purchased software is a cost incurred before or after technological feasibility has been established.

A reporting entity may contract with another entity to develop a custom software product. In this situation, the reporting entity may need to apply judgment to determine whether it is purchasing (a) a completed software product or (b) development services. If the reporting entity is purchasing a completed software product, it would apply the guidance on purchased software to determine the amount of cost to capitalize. If the reporting entity is effectively outsourcing development services, it will need to monitor the development progress to determine when technological feasibility is established. Only fees paid to the third party for development services after technological feasibility is established would qualify for capitalization.

2.5.1 *Purchased software with alternative future use*

If purchased software is acquired before technological feasibility of the software product to be ultimately marketed has been established, but the purchased software has an alternative future use (e.g., the company could alternatively resell the purchased software separately), the cost should be capitalized and accounted for in accordance with its alternative use. However, the amount capitalized would be limited to the amount realizable from the alternative future use. This scenario is described in ASC 985-20-55-14.

ASC 985-20-55-14

An entity may purchase software before technological feasibility has been established. For example, an entity purchases software for $100,000 that can be resold for $75,000. The amount of $25,000 would be charged to research and development, and $75,000 would be capitalized. If the software product reached technological feasibility, the $75,000 would be included in the cost of the software product. If the technological feasibility of the software was never established, the $75,000 would be classified as inventory.

Determining whether purchased software has an alternative future use may require judgment. Chapter 3 of the AICPA *Accounting and Valuation Guide: Assets Acquired to Be Used in Research and Development Activities* includes interpretive guidance to assist in determining whether assets acquired that will be used in R&D activities have an alternative future use.

Excerpt from AICPA Accounting and Valuation Guide Paragraph 3.14

For an asset acquired in an asset acquisition for use in R&D activities to have an alternative future use, the task force believes that (a) it is reasonably expected that the reporting entity will use the asset acquired in the alternative manner and anticipates economic benefit from that alternative use, and (b) the reporting entity's use of the asset acquired is not contingent on further development of the asset subsequent to the acquisition date (that is, the asset can be used in the alternative manner in the condition in which it existed at the acquisition date).

2.6 *Software used as an integral part of a product or process*

If software is embedded in a product or integrated with other non-software components into the ultimate product marketed to customers, ASC 985-20 requires all R&D activities for other components be completed prior to capitalization of software costs, as described in ASC 985-20-25-4.

ASC 985-20-25-4

Software production costs for computer software that is to be used as an integral part of a product or process shall not be capitalized until both of the following conditions have been met:

a. Technological feasibility has been established for the software.

b. All research and development activities for the other components of the product or process have been completed.

For an integrated product, even if technological feasibility of the software has been established, software development costs should be expensed as incurred until all R&D activities for the integrated product are complete. Generally, software is an integral part of a product or process if the software could not be marketed separately from the ultimate product or process sold to customers. For example, software may be developed that will be embedded in a specific piece of hardware. If the software is not marketable as a stand-alone product, software development costs would be

expensed when incurred until all R&D activities for the related hardware in which it will be embedded have been completed.

QUESTION SW 2-2

When should costs be capitalized related to the development of software that will be embedded in a product to be ultimately marketed to customers if the product requires regulatory approval?

PwC response

In accordance with ASC 985-20-25-4, the reporting entity will need to assess whether all R&D activities for the product have been completed prior to capitalizing software development costs, regardless of whether technological feasibility has been established for the software. The reporting entity could have remaining R&D activities to complete leading up to or even after the regulatory approval event. Therefore, judgment may be required to assess when R&D activities are completed. To determine whether R&D activities are completed prior to regulatory approval, the reporting entity should consider whether the approval is a complex process that could require additional R&D or essentially a formality, resulting in no additional R&D.

QUESTION SW 2-3

Does the guidance in ASC 985-20-25-4 also apply to software that has been purchased to be embedded in a product that will be sold to customers?

PwC response

Yes. As discussed in SW 2.5, the cost of purchased software to be externally marketed that has no alternative future use is accounted for in the same manner as costs incurred to develop such software internally. Therefore, if all R&D activities for the other components of the product in which the software will be embedded have not yet been completed, the cost to acquire the software would be expensed as R&D.

2.7 *Amortization of capitalized externally marketed software*

Amortization of capitalized development costs for externally marketed software should commence when the product is available for general release to customers. The amortization of those costs is discussed in ASC 985-20-35-1 through ASC 985-20-35-2.

ASC 985-20-35-1

Capitalized software costs shall be amortized on a product-by-product basis. The annual amortization shall be the greater of the amounts computed using the following:

a. The ratio that current gross revenues for 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 period being reported on.

ASC 985-20-35-2

Because a net realizable value test, which considers future revenues and costs, must be applied to capitalized costs (see paragraph ASC 985-20-35-4), amortization shall be based on estimated future revenues. In recognition of the uncertainties involved in estimating revenue, amortization shall not be less than straight-line amortization over the product's remaining estimated economic life.

The straight-line computation of amortization is the minimum annual amortization expense. Because the guidance requires amortization of the greater of the amounts calculated using the ratio-of-revenues method or the straight-line method, changing between the methods from period to period to meet this requirement is not considered a change in accounting principle. However, if period-to-period comparability is materially impacted, disclosure of the amortization method may be appropriate.

Example SW 2-1 illustrates a scenario in which the reporting entity would conclude that straight-line amortization is appropriate in the first year that it begins amortization.

EXAMPLE SW 2-1

Software amortization example – year 1

Software Corp has capitalized costs associated with software to be sold to customers in accordance with ASC 985-20. At the beginning of 20X1, Software Corp has capitalized $100 million of software costs for Product X. Product X was available for release to customers at the end of 20X0.

Software Corp estimates $625 million of revenue from Product X over its estimated 5-year economic life. Software Corp expects the revenue to be recognized as follows (in millions):

20X1	$100
20X2	$150
20X3	$200
20X4	$100
20X5	$75
Total	**$625**

During 20X1, Software Corp recognizes revenue of $100 million from Product X (which is in line with its expectation). Software Corp's future revenue projections are unchanged.

What is Software Corp's amortization expense for 20X1 related to Product X?

Analysis

Amortization should be the greater of (1) the ratio that current gross revenues for a product bear to the total of current and anticipated future gross revenues for that product or (2) the straight-line method over the remaining estimated economic life of the product including the period being reported on.

Those amounts would be calculated as follows.

- Ratio-of revenues method:

$$\frac{\$100 \text{ million}}{\$100 \text{ million} + \$525 \text{ million}} \times \$100 \text{ million capitalized software} = \$16 \text{ million}$$

- Straight-line method: $100 million / 5 years = $20 million

As a result, during 20X1, Software Corp should record amortization of $20 million for Product X.

Example SW 2-2 illustrates a situation in which a reporting entity needs to adjust the amortization from the straight-line method in Year 1 to the ratio-of-revenues method in Year 2.

EXAMPLE SW 2-2

Software amortization example – year 2

Assume the same facts as Example SW 2-1, including that Software Corp recognized $20 million of amortization for Product X in year 1.

During 20X2, Software Corp exceeds its revenue projections for Product X and records revenue of $325 million and revises its future revenue projections as follows (in millions):

20X1 (actual)	$100
20X2 (actual)	$325
20X3 (projected)	$350
20X4 (projected)	$200
20X5 (projected)	$125
Total	**$1,100**

What is Software Corp's amortization expense for Product X in 20X2?

Analysis

At the beginning of 20X2, Software Corp had unamortized capitalized software of $80 million ($100 million cost less $20 million amortization in 20X1). Amortization should

be the greater of (1) the ratio that current gross revenues for a product bear to the total of current and anticipated future gross revenues for that product or (2) the straight-line method over the remaining estimated economic life of the product including the period being reported on.

Software Corp would determine amortization expense in 20X2 as follows.

- Ratio-of-revenues method:

$$\frac{\$325 \text{ million}}{\$325 \text{ million} + \$675 \text{ million}} \times \$80 \text{ million unamortized software costs} = \$26 \text{ million}$$

- Straight-line method: $80 million / 4 years = $20 million

As a result, during 20X2, Software Corp would record amortization of $26 million for Product X.

Example SW 2-3 illustrates the calculation of software amortization when a change in expected customer demand results in a change in the software's economic life.

EXAMPLE SW 2-3

Software amortization example – year 3

Assume the same facts as Example SW 2-2, including that Software Corp recognized amortization for Project X of $20 million and $26 million in years 1 and 2, respectively.

During 20X3, sales of Product X fall below expectations and amount to only $125 million compared to the projection of $350 million. Software Corp also determines that, as a result of reduced customer demand, the software's 5-year economic life should be reduced to 4 years (or only one additional remaining year beyond 20X3). Software Corp revises its future revenue projections as follows (in millions):

20X4	$200
Total	**$200**

What is Software Corp's amortization expense in 20X3 related to Product X?

Analysis

At the beginning of 20X3, Software Corp had unamortized software capitalized cost of $54 million ($100 million less cumulative amortization of $46 million in 20X1 and 20X2). Amortization should be the greater of (1) the ratio that current gross revenues for a product bear to the total of current and anticipated future gross revenues for that product or (2) the straight-line method over the remaining estimated economic life of the product including the period being reported on.

Software Corp would determine amortization expense in 20X3 as follows.

- Ratio-of-revenues method:

$$\frac{\$125 \text{ million}}{\$125 \text{ million} + \$200 \text{ million}} \times \$54 \text{ million unamortized software costs} = \$21 \text{ million}$$

- Straight-line method: $54 million / 2-year adjusted remaining economic life = $27 million

As a result, during 20X3, Software Corp would record amortization of $27 million for Product X.

2.8 *Impairment of capitalized externally marketed software*

The impairment test for externally marketed software is based on net realizable value (NRV) as described in ASC 985-20-35-4, which is similar to the NRV concept for inventory in ASC 330.

ASC 985-20-35-4

At each balance sheet date, the unamortized capitalized costs of a computer software product shall be compared to the net realizable value of that product. The amount by which the unamortized capitalized costs of a computer software product exceed the net realizable value of that asset shall be written off. 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 responsibility set forth at the time of sale. The reduced amount of capitalized computer software costs that have been written down to net realizable value at the close of an annual fiscal period shall be considered to be the cost for subsequent accounting purposes, and the amount of the write-down shall not be subsequently restored.

The NRV test is required to be performed at each balance sheet date. ASC 985-20 indicates that the unamortized capitalized costs to be used for subsequent accounting purposes (i.e., subsequent NRV tests) is the amount determined by the NRV test performed at the close of the previous annual fiscal period.

QUESTION SW 2-4

In performing the NRV test, should estimated future revenues include both revenues from the sale of the software and software-related revenue (e.g., post-contract customer support (PCS) or other related services)?

PwC response

Yes. ASC 985-20-35-4 states that the future gross revenues from the software product should be used in applying the NRV test. The product's future revenues should be reduced by the estimated future costs of completing and disposing of the product, including maintenance and other PCS.

PCS revenues are often intended to recover not only capitalized costs, but also to fund the future development of upgrades and enhancements covered by the PCS arrangement. In applying the NRV test, it would be appropriate to include PCS revenues; however, those revenues should be reduced by the expected future costs of completing the upgrades or enhancements covered by the PCS arrangement.

Future costs should include all costs that will be incurred to support the PCS arrangement and the costs of any upgrades that will be provided to the customer.

Similarly, the reporting entity may also use the same software in hosting arrangements with customers (e.g., SaaS). In that situation, the related service revenues and costs should be included in the NRV test.

QUESTION SW 2-5

In performing the NRV test, should the estimated future gross revenues be discounted to their present value?

PwC response

No. ASC 985-20 does not discuss the discounting of gross revenues. Conceptually, costs of externally marketed software are similar to inventory costs. ASC 330 does not discuss the discounting of inventory when performing the NRV test. As such, the NRV test for externally marketed software should be performed on an undiscounted basis.

Example SW 2-4 illustrates a NRV assessment performed on capitalized costs related to externally marketed software.

EXAMPLE SW 2-4

NRV assessment for externally marketed software

Tech Corp has capitalized costs related to the development of software that it plans to market externally. The unamortized cost of the software at the end of 20X1 is $100 million. The remaining costs to be incurred and capitalized during 20X2, before general release, are $10 million. In summary, Tech Corp expects the following estimated future revenue and costs (in millions):

	Estimates as of 20X1	Estimates as of 20X2
Total estimated future revenue	$220	$200
Total cost to fulfill the above revenues	$80	$100
Costs to complete the software product	$10	$0

What are the results of the Tech Corp's assessment of the NRV of software costs at the end of each fiscal period?

Analysis

Tech Corp would assess the NRV of the software costs as follows:

	Estimates as of 20X1	Estimates as of 20X2
Unamortized software costs	$100	$110
Net realizable value:		
Total estimated future revenue	$220	$200
Less: costs to complete	($10)	--
Less: costs to fulfill revenues	($80)	($100)
	$130	$100
Excess / (impairment)	$30	($10)

At the end of 20X1, based on the then-current estimates of costs to complete and expected future revenues, no impairment is indicated.

At the end of 20X2, as a result of a lower revenue forecast and higher costs to fulfill the revenues, the unamortized capitalized costs exceed their NRV by $10 million; therefore, an impairment of $10 million would be recognized. The updated unamortized software costs at the end of 20X2 would be $100 million.

2.8.1 *Costs of abandoned software development projects*

If costs have been capitalized for a software product that has achieved technological feasibility, but it subsequently becomes no longer probable that the software product will be completed, capitalized costs should be written down to the lower of cost or fair value less cost to sell. If the software will be completed, but will not include all of the features included in the original product design, the lower of cost or fair value model does not need to be applied as long as the product will continue to be saleable without the omitted features. However, the capitalized costs would still be subject to a NRV test, which may require write-down based on changes in projected revenue as a result of the modified features.

2.9 Costs incurred after general release to customers

Costs incurred after general release of software to customers may relate to (a) product enhancements or (b) maintenance and customer support. Product enhancements are included in the scope of ASC 985-20 and may require capitalization, while maintenance and customer support costs are expensed when incurred.

Judgment may be required to distinguish between costs that relate to product enhancements and costs that relate to maintenance. This assessment is similar to that used for capital improvements. In general, costs that extend the life or improve the marketability of the original software product or create a new revenue stream may qualify for capitalization as a product enhancement. In contrast, costs to maintain the promised functionality of existing products are likely maintenance costs. For example, if a reporting entity commits to maintaining compatibility between its software and related hardware, the costs of doing so would generally meet the definition of

maintenance. However, if an existing product is modified to run on a different type of hardware, thereby expanding the market for the product, this activity would generally be classified as product enhancement.

Similar to other software development costs, costs relating to product enhancement should be expensed when incurred as R&D until the technological feasibility of the enhancement is established. Once technological feasibility is established, capitalization and amortization of the product enhancement costs over the estimated life of the enhancement would be required.

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

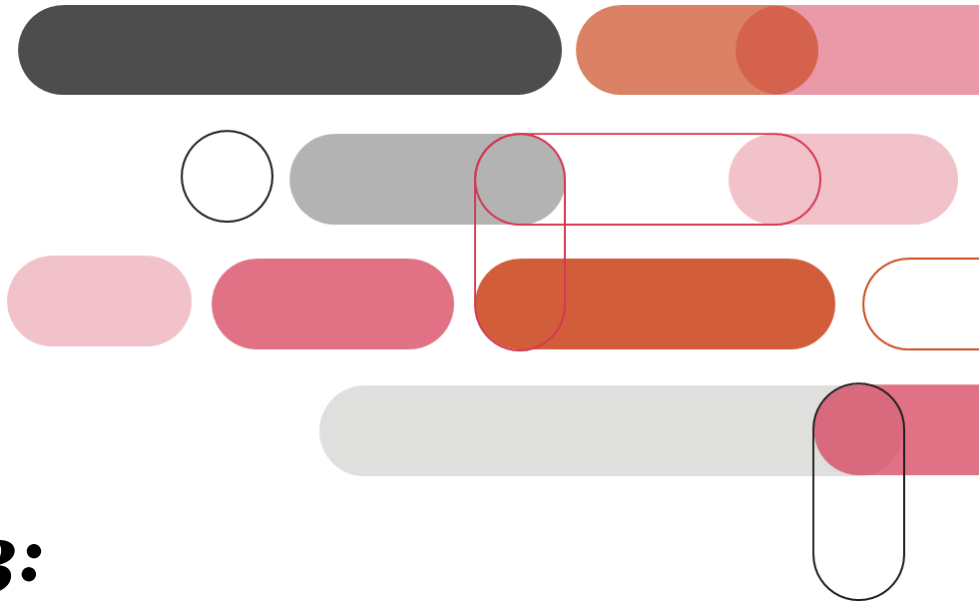
2.9.1 Amortization of software product enhancement costs

Once a product enhancement has been completed, the amortization of the capitalized costs depends on whether the original product will continue to be separately marketed.

If the original product will no longer be separately marketed, any unamortized cost of the original product should be included with the cost of the enhancement for purposes of applying the NRV test and amortization provisions. Under this method (sometimes referred to as the “carryover method”), the combined amount is amortized over the useful life of enhanced product. The estimated useful life of the enhanced product will likely extend beyond the useful life initially determined for the original product.

If the original product will remain on the market along with the enhanced version, an allocation of the unamortized cost of the original product between the original product and the enhanced version will be necessary. Under this method (sometimes referred to as the “vintage method”), the costs of the initial product and the product enhancement are amortized separately. The initial software would continue to be amortized over its remaining useful life while the costs of the enhancement would be separately tracked and amortized over the remaining useful life of the enhanced version.

Refer to SW 2.7 for guidance on the amortization of capitalized externally marketed software costs.



Chapter 3: Internal-use software—updated December 2021

3.1 Internal-use software—chapter overview—updated April 2025

ASC 350-40 provides the guidance for the costs to develop or obtain software for internal use. That guidance is similar to the guidance for the costs of acquiring other long-lived assets with respect to which costs are capitalized and how the costs are subsequently amortized and tested for impairment.

A software license purchased for internal use should be accounted for as the acquisition of an intangible asset. To the extent any or all of the software licensing fees are still payable on the acquisition date of the license, a liability would be recognized for those payments. The intangible software asset should be recognized and measured in accordance with ASC 350-30, *Intangibles—Goodwill and Other—General Intangibles Other than Goodwill*. See BCG 8.

If a reporting entity is developing, modifying, or implementing software for internal use, the assessment of whether costs should be expensed or capitalized depends on the project stage during which the costs are incurred. The guidance defines three stages in the development of internal-use software:

- Preliminary project stage (see SW 3.2)
- Application development stage (see SW 3.3)
- Postimplementation-operation stage (see SW 3.4)

Generally, only costs incurred during the application development stage are eligible for capitalization. See SW 3.2 through SW 3.4 for a description of each project stage and examples of activities associated with each stage.

Figure SW 3-1 illustrates an example project timeline for the development of internal-use software.

FIGURE SW 3-1
Internal-use software – example development timeline

* Refer to SW 3.3 for requirements on commencing capitalization of qualifying costs.

** Certain costs are expensed as incurred, such as training costs and data conversion costs.

Note about ongoing standard setting

The FASB has an active project related to the accounting and disclosure of software costs. Financial statement preparers and other users of this publication are therefore encouraged to monitor the status of the project, and if finalized, evaluate the effective date of the new guidance and the implications on the accounting and related disclosures.

3.2 Preliminary project stage

The first stage of development described in ASC 350-40-25 is the preliminary project stage.

Definition from ASC Master Glossary

Preliminary Project Stage: When a computer software project is in the preliminary project stage, entities will likely do the following:

a. Make strategic decisions to allocate resources between alternative projects at a given point in time. For example, should programmers develop a new payroll system or direct their efforts toward correcting existing problems in an operating payroll system?

b. Determine the performance requirements (that is, what it is that they need the software to do) and systems requirements for the computer software project it has proposed to undertake.

c. Invite vendors to perform demonstrations of how their software will fulfill an entity's needs.

d. Explore alternative means of achieving specified performance requirements. For example, should an entity make or buy the software? Should the software run on a mainframe or a client server system?

e. Determine that the technology needed to achieve performance requirements exists.

f. Select a vendor if an entity chooses to obtain software.

g. Select a consultant to assist in the development or installation of the software.

The following activities are generally part of the preliminary project stage:

- Conceptual formulation of ideas and alternatives
- Evaluation of alternatives
- Determination of existence of needed technology
- Final selection of alternatives

These costs are generally incurred in the early stages of a project when the reporting entity is exploring its technological needs and exploring various alternatives.

Internal and external costs incurred during the preliminary project stage should be expensed as incurred.

3.3 Application development stage

The next stage of development described in ASC 350-40-25 is the application development stage. Capitalization of qualifying costs during the application development stage should begin when both of the following occur:

- The preliminary project stage is completed, and

- Management, with the relevant authority, implicitly or explicitly authorizes and commits to funding a project and it is probable that the project will be completed, and the software will be used to perform the function intended (e.g., execution of a contract with a third party, approval of expenditures related to internal development, or a commitment to obtain software from a third party).

The following activities are generally part of the application development stage:

- Design of chosen path, including software configuration and interfaces
- Coding
- Installation of hardware
- Testing, including parallel processing

Capitalization should cease no later than the point at which a software project is substantially completed and ready for its intended use. Software is ready for its intended use after all substantial testing is completed. This may occur before the software is placed in service.

If, prior to completion, it becomes not probable that a software project will be completed and placed into service, a reporting entity should cease capitalization of costs and assess the in-process software asset for impairment. Refer to SW 3.8 for additional discussion on the impairment of capitalized internal-use software.

3.3.1 Capitalizable costs for internal-use software—updated April 2025

During the application development stage, some costs are capitalized while other costs are expensed as incurred. In general, costs that are directly attributable to the development of the software are capitalized, while indirect costs are expensed as incurred.

ASC 350-40-30-1 specifies the types of direct costs that should be capitalized.

ASC 350-40-30-1

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

a. External direct costs of materials and services consumed in developing or obtaining internal-use computer software. Examples of those costs include but are not limited to the following:

   1. Fees paid to third parties for services provided to develop the software during the application development stage
   2. Costs incurred to obtain computer software from third parties
   3. Travel expenses incurred by employees in their duties directly associated with developing software.

b. Payroll and payroll-related costs (for example, costs of employee benefits) for employees who are directly associated with and who devote time to the internal-use computer software project, to the extent of the time spent directly

on the project. Examples of employee activities include but are not limited to coding and testing during the application development stage.

c. Interest costs incurred while developing internal-use computer software. Interest shall be capitalized in accordance with the provisions of Subtopic 835-20.

Stock-based compensation costs should be included in the same line or lines as the cash compensation (e.g., payroll costs) paid to the employees receiving the stock-based awards. Therefore, payroll-related costs described in ASC 350-40-30-1(b) above include stock-based compensation for employees who are directly associated with and who devote time to the software project.

The following types of costs are expensed as incurred, even during the application development stage:

- Training costs—training of personnel on how to use the new, internally developed software application

  The costs to "train" an AI algorithm are subject to evaluation under the general capitalization framework and not subject to the guidance for training costs.

- Data conversion costs (such as employee time spent physically converting data), except for costs to develop or obtain software that allows for access or conversion of old data by new systems

  The process of data conversion from an old system to a new one may include purging or cleansing existing data, reconciling the data in the old and new systems, creating new or additional data, and converting old data to the new system.

- General and administrative costs and overhead costs

3.3.1.1 Artificial Intelligence models

There are two specific types of costs that are prevalent in AI software development that would need to be evaluated to determine if they are considered a type of cost eligible for capitalization:

- Costs to develop the algorithm
- Cost of data purchased to train the model or facilitate machine learning

Costs to develop the algorithm

A key component of an AI software program is the algorithm that provides the key conceptual functionality of the software program. Broadly speaking, an algorithm is a procedure for solving a problem (commonly a mathematical problem) in a finite number of steps, often involving repetition of an operation. The problem-solving procedure is coded into an AI software program and becomes the core functional engine of that AI program. The cost of coding of an algorithm into a software program is assessed and subject to capitalization in the same way as the coding effort in any other software development project. The cost of coding the algorithm into the software program is generally either a fee paid to a third party to perform the coding or a payroll-related cost related to an employee who performs the coding. In either case, the coding cost is capitalizable as a direct cost of developing the software. Similarly, when an AI-enabled machine is leveraged in the development of the

algorithm, the compute cost (including people, equipment, cloud services costs, etc.) is also eligible for capitalization.

The more difficult judgments relate to determining whether costs related to activities other than software coding are capitalizable. In some cases, algorithms are explored and developed by experts (e.g., mathematicians, academic researchers, engineers) before the algorithms are coded into the software. In other cases, the software engineers (either alone or working with others) develop the algorithm as they code the software. The key factor in determining the appropriate accounting for the costs of non-coding activities associated with developing algorithms is whether those activities are (1) directly related to development of the software's algorithm (and capitalized under ASC 350-40-30-1(a) or ASC 350-40-30-1(b)) or (2) essentially research and development expenses incurred in creating a separate internally developed intangible asset apart from the software (and expensed under ASC 730-10). Judgment will be required to make this determination.

Cost of purchased data

A significant input into the development of AI software is data. For AI software to produce relevant and accurate results, the software must interact with large quantities of data relevant to its use case. For example, an AI software program that helps to perform medical diagnoses would need health-related data such as patient profiles, disease symptoms, and environmental factors. The more data on which the AI software can be trained or that the AI software engine can ingest, the greater the accuracy of the output. As such, reporting entities developing AI software often purchase data from third parties during the development of the software and for ongoing updates and enhancements over the life of the software platform. The process of utilizing data in the development of an AI program is often referred to as "training" the AI model. The concept of training the AI model is different than training people to use the software, the costs of which are explicitly required to be expensed as incurred under ASC 350-40-25-4.

Purchased data is not software. Rather, it is an intangible asset subject to ASC 350-30. When a software development project includes the purchase of intangible assets other than software (such as data) as an input, the key factor in determining the accounting for the cost of the purchased intangible asset is whether the purchased intangible asset has an alternative future use in other research and development projects, other software development projects, or otherwise.

- When the purchaser of the data can use the data to train multiple AI models (software programs), the purchased data is considered to have an alternative future use. ASC 730-10-25-2(c) requires the cost of intangibles assets that are purchased for use in research and development activities and have alternative future uses to be accounted for in accordance with the intangible asset guidance in ASC 350.

- The cost of an intangible asset purchased for use in research and development activities that have no alternative future uses are expensed as incurred. Similarly, we believe that an intangible asset purchased for use in a software development project that has no alternative future use is considered a software development cost subject to the guidance in ASC 350-40 or ASC 985. The cost of purchased data does not meet the definition of a capitalizable cost in ASC 350-40 because it is not a material or service consumed in the development of the software.

We expect that in many cases purchased data will have an alternative future use and, therefore, will be subject to the intangible asset guidance in ASC 350. Accordingly, in these cases, the cost of the purchased data will be capitalized as an

intangible asset and amortized over its useful life, as determined in accordance with ASC 350-30-35-2. Amortization of the intangible asset will be recognized as a cost of whatever activity the asset is used in – if in the development of a software cost, then subject to the software cost guidance. Said differently, once capitalized as an asset, the cost of using the asset (amortization) should be evaluated under the research and development or software development cost guidance.

3.4 Postimplementation-operation stage

The postimplementation-operation stage begins when the internal-use software is ready for its intended use. During this stage, all training and routine maintenance costs should be expensed as incurred.

3.4.1 Internal-use software upgrades and enhancements

The accounting for upgrades and enhancements to internal-use software follows the same accounting model as other internal-use software costs (i.e., qualifying costs are capitalized during the application development stage), provided it is probable that the expenditures will result in additional functionality. Additional functionality means that the software modifications enable the software to perform tasks that it previously was not capable of performing.

Judgment may be required to assess whether upgrades and enhancements result in additional functionality. Examples of modifications that result in additional functionality include modifications that allow software to operate on an additional hardware platform or in an additional environment. ASC 350-40-25-10 indicates that if reporting entities cannot reasonably separate costs of relatively minor upgrades and enhancements from costs of maintenance, the costs of those minor upgrades and enhancements should be expensed as incurred.

QUESTION SW 3-1

How should reporting entities account for software upgrades and enhancements to purchased software that are received through a post-contract customer support (PCS) arrangement?

PwC response

In connection with the purchase of software from a third party, reporting entities often also contract for PCS related to the software, which includes the right to receive unspecified when-and-if available updates and upgrades. Pursuant to ASC 350-40-25-11, unspecified updates and upgrades received through a PCS arrangement should be expensed over the contract period on a straight-line basis, unless another systematic and rational basis is more representative of the pattern in which the services are received. Costs related to specified upgrades received by the reporting entity should be assessed under the general guidance for software upgrades and enhancements to determine what costs, if any, should be capitalized (e.g., based on whether they result in additional functionality). It may be necessary to allocate costs paid to a third-party vendor between costs that are capitalizable and costs that must be expensed if an arrangement includes multiple elements, as discussed in SW 3.6.

3.5 Agile software development for internal-use software

One of the key challenges in accounting for software development costs stems from the continuous technological changes in software development practices, which now largely follow agile development principles. An agile development process uses sprints for planning and execution in which a larger project is typically broken down into smaller increments and feedback is used to continuously iterate the development process, resulting in features and functionalities that are individually developed and

continuously changing. Historically, reporting entities may have utilized a linear or "waterfall" method that involved a sequential software design process that "flowed" through the development stages contemplated in ASC 350-40 (see SW 3.1). An agile or iterative software development approach may not follow the distinct project stages contemplated in ASC 350-40. In this situation, reporting entities should apply the guidance based on the nature of the costs incurred. For example, costs incurred for software coding are generally capitalized, while those incurred for training activities should be expensed as incurred. Judgment may be required to appropriately track and categorize costs based on the activities being performed (e.g., planning, coding, testing, training) to identify costs that should be capitalized.

3.6 Multiple-element software arrangements

A reporting entity may enter into a contract with a third party that includes multiple elements, such as a software license, implementation services, hosting services, and training. The consideration paid to the third party should be allocated to each element based on its relative standalone price. After allocation, the accounting for each element (e.g., whether the costs should be capitalized or expensed) will depend on the nature of the cost incurred.

ASC 350-40-30-4 provides guidance on allocating costs to multiple elements in an arrangement.

ASC 350-40-30-4

Entities may purchase internal-use computer software from a third party or may enter into a hosting arrangement. In some cases, the price includes multiple elements, such as 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. Entities shall allocate the cost among all individual elements. The allocation shall be based on the relative standalone price of the elements in the contract, not necessarily separate prices stated within the contract for each element. Those elements included in the scope of this Subtopic shall be accounted for in accordance with the provisions of this Subtopic.

The concept of "relative standalone price" is similar to the concept of "relative standalone selling price" described in ASC 606 (see RR 5.2 and RR 5.3). It represents the price at which a reporting entity would purchase an element of a contract separately. Determining the relative standalone price of the various elements in the contract may require the use of estimates. Management should consider all relevant information, such as information from the negotiation process with the vendor, in estimating the standalone price. A reporting entity should not assume that the price stated within the contract represents the standalone price.

In some arrangements, a reporting entity will pay a third party over time (e.g., a recurring monthly payment) for multiple elements, which could include a software license or implementation services for a cloud-computing arrangement (CCA, see SW 4) that require capitalization at or near the beginning of the arrangement. In this situation, the reporting entity should allocate the total fee to all of the elements and should capitalize the full cost of the license or implementation services received, with a liability for any amounts not yet paid based on the terms of the contract. Said differently, regardless of the timing of payment, an asset is recognized for those software costs that are capitalizable when the related licenses or services are received.

Example SW 3-1 and Example SW 3-2 illustrate the accounting for a multiple-element software arrangement.

EXAMPLE SW 3-1

Allocating costs for a multiple-element software arrangement (on-premises software)

Data Analytics Co enters into an agreement with Software Co to license on-premises data analytics software. The contract also includes routine maintenance of the software and training for employees of Data Analytics Co. Additionally, Data Analytics Co engages the same vendor to perform data conversion and configuration services as part of implementing the new software. The total contract price is $10 million, payable at contract inception.

Software Co offers the software and maintenance services separately for $7 million and $1 million, respectively. Software Co does not offer the training, data conversion, or configuration services separately; however, Data Analytics Co obtained information about pricing from other vendors in the vendor selection process. Data Analytics Co uses this information to estimate standalone prices for the training, data conversion, and configuration of $0.5 million, $2 million, and $1.5 million, respectively.

How should Data Analytics Co account for each element in the agreement?

Analysis

Data Analytics Co should allocate the contract price of $10 million to each of the elements in the contract based on their relative standalone price.

Element	Standalone price	Allocation %*	Allocated amount
Software license	$7,000,000	58%	$5,800,000
Maintenance	1,000,000	8%	800,000
Training	500,000	4%	400,000
Data conversion	2,000,000	17%	1,700,000
Configuration	1,500,000	13%	1,300,000
	$12,000,000	100%	$10,000,000

* The allocation percentages have been rounded for presentation purposes.

The amounts allocated to the on-premises software license, as well as the configuration services, should be capitalized in accordance with internal-use software guidance. The amount allocated to maintenance, training, and data conversion services should be expensed as incurred. Because Data Analytics Co pays the total contract price upfront, any amounts prepaid for these services should be initially recognized as a prepaid expense.

EXAMPLE SW 3-2

Allocating costs for a multiple-element software arrangement (on-premises software and software as a service)

Data Cloud Co enters into a three-year noncancellable agreement with Computing Co to license on-premises data analytics software and to access Computing Co's cloud-based customer relationship management (CRM) platform (a SaaS arrangement). The agreement does not give Data Cloud Co the right to take possession of the CRM software. Additionally, Computing Co will provide implementation services to customize and configure the SaaS offering. The total

contract price is $9 million, with equal monthly payments of $250,000 over the three-year contract period.

Computing Co offers the on-premises software and SaaS separately for $5 million and $6 million, respectively. Computing Co does not offer the implementation services separately; however, Data Cloud Co obtained information about pricing from other vendors in the vendor selection process. Data Cloud Co uses this information to estimate a standalone price of $1 million for the implementation services.

How should Data Analytics Co account for each element in the agreement?

Analysis

Data Cloud Co should allocate the contract price of $9 million to each of the elements in the contract based on their relative standalone price.

Element	Standalone price	Allocation %*	Allocated amount
Software license	$5,000,000	42%	$3,780,000
SaaS	6,000,000	50%	4,500,000
Implementation services	1,000,000	8%	720,000
	$12,000,000	100%	$9,000,000

*The allocation percentages have been rounded for presentation purposes.

The amount allocated to the on-premises software license and the SaaS implementation services (refer to SW 4) should be capitalized in accordance with internal-use software guidance. The amount allocated to the SaaS should be expensed over the contract term. Because the contract price will be paid monthly over the three-year contract, Data Cloud Co will initially record a liability for the amounts allocated to the software license and the capitalizable implementation services as they are received. For example, assuming the software license and implementation services are received in the first month, and the $250,000 billing occurs at the end of the month, Data Cloud Co would record the following entry (excluding the amortization of the software license and the capitalized SaaS implementation costs):

Internal-use software	3,780,000	
Capitalized implementation costs	720,000	
Operating expense	125,000*	
Accounts payable (billed)		250,000
Accrued expenses (unbilled)		4,375,000

* $125,000 (one month of SaaS) = $4,500,000 (SaaS) / 36 months (noncancellable term of the SaaS agreement)

3.7 Amortization of capitalized internal-use software costs

Capitalized internal-use software costs are amortized over the estimated useful life of the software, generally on a straight-line basis, unless another systematic and

rational basis is more representative of the software's use. ASC 350-40-35-5 provides the factors to consider in determining the appropriate life.

ASC 350-40-35-5

In determining and periodically reassessing the estimated useful life over which the costs incurred for internal-use computer software will be amortized, entities shall consider the effects of all of the following:

a. Obsolescence

b. Technology

c. Competition

d. Other economic factors

e. Rapid changes that may be occurring in the development of software products, software operating systems, or computer hardware and whether management intends to replace any technologically inferior software or hardware.

Given the history of rapid changes in technology, software often has had a relatively short useful life.

Amortization of internal-use software should begin when the software is ready for its intended use, regardless of whether the software has actually been placed in service. As discussed in SW 3.3, software is ready for its intended use after all substantial testing is completed.

Commencement of amortization should be assessed at the module or component level. ASC 350-40-15-2 provides an example of an accounting software system that contains separate modules, including a general ledger, an accounts payable subledger, and an accounts receivable subledger. In this example, each element might be viewed as a module of the entire accounting software system.

When the functionality of a software module is entirely dependent on the completion of other modules (that are not yet designed, but for which completion is probable), amortization should not begin until all of the modules on which functionality is dependent are ready for their intended use.

QUESTION SW 3-2

Company A begins to use a software module that it developed and which is functional on a standalone basis. Company A plans to develop four additional modules that will provide additional functionality to the software. When should amortization begin on the developed software module?

PwC response

Because the initial software module has standalone functionality that is not dependent on the completion of the other modules, amortization should begin when the initial software module is completed and ready for its intended use.

3.8 *Impairment of capitalized internal-use software costs*

Internal-use software assets generally should be tested for impairment as part of the related asset group in accordance with the guidance in ASC 360, *Property, Plant, and Equipment*, related to the impairment of long-lived assets. This guidance applies to software that has been developed (or is probable of being completed). See PPE 5 for more information on long-lived asset impairments.

In order to assess long-lived assets for impairment, assets are required to be grouped at the lowest level for which there are identifiable cash flows that are largely independent of the cash flows from other groups of assets (i.e., the asset group level). Internal-use software should be assigned to the applicable asset group when the reporting entity performs its long-lived asset impairment tests. See PPE 5.2.1 for details regarding the determination of the asset group.

If a reporting entity commits to a plan to cease use of its internal-use software assets before the end of its previously estimated useful life, those assets should be accounted for in accordance with long-lived assets to be abandoned guidance in ASC 360-10-35-47 and ASC 360-10-35-48. Accordingly, even if the broader asset group does not fail the recoverability test described in ASC 360, the reporting entity should revise the estimated useful life of the internal-use software asset and related amortization in accordance with ASC 250.

Impairment testing is performed when a triggering event has been identified for the asset group or for an individual asset included in the broader asset group (if significant). ASC 360-10-35-21 provides examples of when to test long-lived assets for recoverability and impairment. For further discussion regarding long-lived asset impairment triggers, see PPE 5.2.3. ASC 350-40-35-1 includes examples of triggering events for capitalized software. When a reporting entity identifies an impairment indicator for capitalized software included in a broader asset group, the reporting entity should consider the significance of the asset in relation to the overall asset group and the facts and circumstances surrounding the impairment indicator to determine whether impairment testing at the asset group level may be required. See PPE 5.2.3.1. These impairment indicators are examples and should not be considered the only potential indicators that an asset or an asset group may not be recoverable.

Excerpt from ASC 350-40-35-1

The guidance is applicable, 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:

a. Internal-use computer software is not expected to provide substantive service potential.

b. A significant change occurs in the extent or manner in which the software is used or is expected to be used.

c. A significant change is made or will be made to the software program.

d. Costs of developing or modifying internal-use computer software significantly exceed the amount originally expected to develop or modify the software.

3.8.1 *Internal-use software not probable of completion*

Even when the carrying amount of an asset group containing internal-use software is deemed to be recoverable, capitalized software may need to be impaired if it is no longer probable that the software being developed will be completed. This guidance differs from the model utilized when it remains probable that the software being developed will be completed and placed into service. ASC 350-40-35-3 discusses the accounting when development of the software is no longer probable.

ASC 350-40-35-3

When it is no longer probable that computer software being developed will be completed and placed in service, the asset shall be reported at the lower of the carrying amount or fair value, if any, less costs to sell. The rebuttable presumption is that such uncompleted software has a fair value of zero. Indications that the software may no longer be expected to be completed and placed in service include the following:

a. A lack of expenditures budgeted or incurred for the project.

b. Programming difficulties that cannot be resolved on a timely basis.

c. Significant cost overruns.

d. 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.

e. 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.

f. Business segment or unit to which the software relates is unprofitable or has been or will be discontinued.

As indicated in the above guidance, software being developed that is no longer probable of completion should be reported at the lower of cost or fair value less cost to sell. ASC 350-40-35-3 stipulates that there is a rebuttable presumption that uncompleted software has no value. The assessment of impairment for uncompleted software is performed at the module or component level.

3.9 *Internal-use software subsequently marketed*

As discussed in SW 1.4, a reporting entity accounts for software costs in accordance with the guidance on internal-use software in ASC 350-40 when no substantive plan exists or is being developed to market the software externally. In some circumstances, a reporting entity that previously had no plan to market software to other parties—and therefore, capitalized costs under ASC 350-40—will subsequently decide to license or sell the software. In this situation, ASC 350-40-35-7 requires the proceeds received from the license or sale of the software, net of direct incremental costs of marketing (e.g., commissions, software reproduction costs, warranty and service obligations, installation costs), to be applied first against the carrying value of any capitalized internal-use software costs. No profit should be recognized until aggregate proceeds from the licenses and amortization have reduced the carrying amount of the software to zero. Once the carrying amount has been reduced to zero,

any additional proceeds should be recognized in accordance with ASC 606—if licensing or selling software is part of the reporting entity's ongoing major or central operations—or ASC 610-20, *Other Income-Gains and Losses from the Derecognition of Nonfinancial Assets*.

If during the development of internal-use software, a reporting entity decides to market the software to others, the guidance in ASC 985-20 for externally marketed software should be followed on a prospective basis. As discussed in SW 1.4, if a reporting entity has a pattern of selling software to third parties that was originally being developed to use internally, there is a rebuttable presumption that any software developed by that reporting entity is intended for sale, lease, or other marketing.

See SW 1.5.2 for discussion of arrangements when a reporting entity provides access to software through a cloud computing arrangement.

Example SW 3-3 illustrates the accounting for the subsequent license of internal-use software to other parties.

EXAMPLE SW 3-3

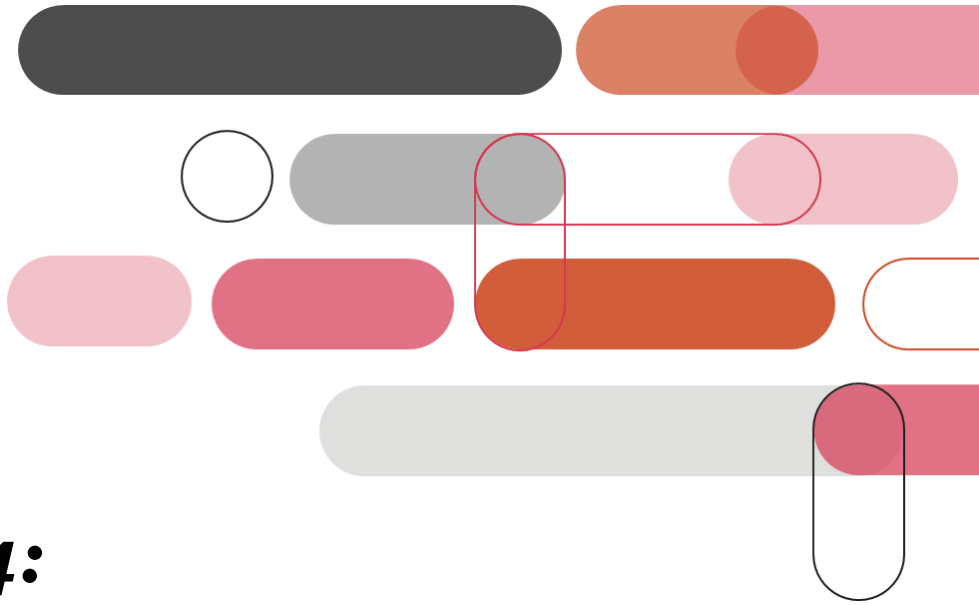
Accounting for the license of internal-use software to other parties

Retail Co is a national retail enterprise that has agreed to sell its stores located in the northeast region to a third-party purchaser. As part of this sale, the purchaser will license a software package that Retail Co had previously developed for its internal use. Retail Co had never intended to market or license the software externally; therefore, the software development costs were capitalized based on the guidance for internal-use software. To facilitate the transfer of the purchased operations, Retail Co has agreed to license the software to the purchaser for one year at a market rate of $100,000. The carrying value of the internally developed software is $5,000.

How should Retail Co account for the license of its software to the purchaser?

Analysis

The $100,000 received from the license of the software should be applied first against the carrying value of the capitalized internal-use software development costs ($5,000). The remaining $95,000 should be recognized as either revenue or a gain, depending on whether licensing software is part of Retail Co's ongoing major or central operations. In this context, it would likely be most appropriate for Retail Co to recognize the remaining $95,000 as a gain outside of revenue.



Chapter 4: Cloud computing implementation costs—updated December 2021

4.1 Cloud computing implementation costs—chapter overview

As companies move their data, applications, and platforms to the cloud, software that a customer would have traditionally installed locally on its own servers is now often hosted on a vendor's (or a third-party cloud platform provider's) servers and accessed by the customer remotely. These arrangements are sometimes referred to as hosting arrangements or cloud computing arrangements (CCAs). Examples of CCAs include software as a service and other "as a service" arrangements, including platform as a service and infrastructure as a service. Some CCAs transfer a license to the software in addition to the service of hosting the software. The assessment of whether a CCA includes a license is discussed in SW 1.5.

A reporting entity that purchases services through a CCA provided by a third-party vendor may be entering into an arrangement that is solely a service contract. Although the reporting entity is not acquiring software in this situation, the determination of whether to capitalize the related implementation costs is based on the internal-use software guidance in ASC 350-40, as further discussed in this chapter. The same model is used because the types of activities and the nature of the costs to implement on-premises software and a CCA can be similar. However, presentation of costs related to a software license and a CCA that is a service will differ. In essence, purchasing a software asset is an investment in a long-lived asset; purchasing services through a CCA is an operating expense. These presentation differences are discussed in detail in FSP 8.6.

A reporting entity that is the vendor providing a CCA to customers will need to assess whether the arrangement includes a license (i.e., whether the software is transferred to the customer) to determine which guidance to apply to the related software development costs. The scoping considerations and accounting implications for vendors providing a CCA are discussed in SW 1.5.2.

4.2 Implementation costs of a CCA

Although the reporting entity does not acquire software in a CCA that is solely a service contract, costs to implement the CCA are accounted for following the same model as the guidance for internal-use software in ASC 350-40, which is described in SW 3. Accordingly, whether implementation costs are capitalized or expensed depends on the nature of the costs and the project stage during which they are incurred. Because the guidance in ASC 350-40 generally addresses the development of software, applying the guidance to implementation activities for a CCA may require judgment.

As discussed in SW 3, ASC 350-40 describes three project stages:

- **Preliminary project stage:** For a CCA, this stage generally consists of planning activities; during this stage, costs are expensed as incurred.
- **Application development stage:** For a CCA, this stage generally consists of configuration and customization activities; during this stage, certain costs are capitalized (see further discussion below).
- **Postimplementation-operation stage:** For a CCA, this stage generally consists of maintenance activities; during this stage, costs are expensed as incurred.

Consistent with the internal-use software guidance, in general:

- costs related to coding and testing activities during the application development stage are capitalized (e.g., costs for configuration and customization),
- costs related to training activities are expensed as incurred, and
- costs related to data conversion activities are expensed as incurred.

4.2.1 *Capitalizable CCA implementation costs*

Both internal and external costs incurred during the application development stage are eligible for capitalization. Examples include the following:

- Payroll and payroll-related costs (e.g., costs of employee benefits or stock-based compensation) for employees who are directly associated with and who devote time to implementation
- Fees paid to the CCA service provider or other third-party vendors providing implementation services. These may also include travel expenses directly associated with providing implementation services.

4.2.2 *Accounting for CCA service fees*

The accounting for the costs for the CCA service itself is not specifically addressed in ASC 350-40. The reporting entity should account for these costs similar to costs for other service contracts, which are generally expensed as incurred or as the services are received. Payments made in advance of the CCA service being received should generally be recorded as a prepaid expense.

Often, the reporting entity will begin incurring CCA service fees prior to completion of implementation activities (e.g., in order to migrate data from the existing system to new cloud-based solution). The CCA service fees incurred during this period should generally not be capitalized as an implementation activity, even if they are incurred prior to completion of implementation.

CCAs with third-party service providers often include multiple elements (e.g., hosting service, implementation costs); therefore, total consideration paid to the CCA service provider may need to be allocated to the different elements as discussed in SW 3.6.

4.3 *Amortization of capitalized CCA implementation costs*

Capitalized CCA implementation costs should be amortized over the term of the related CCA. Amortization expense should be recognized on a straight-line basis, unless another systematic and rational basis is more representative of the pattern in which the reporting entity expects to benefit from its right to access to the hosted software. The pattern of amortization should not be based on expectations about the reporting entity's usage of the hosted software (e.g., how many transactions the reporting entity will process or how many users will access the hosted software).

The term of the service for amortization purposes should include the initial noncancelable service term and all of the following:

- Periods covered by an option to extend if the reporting entity is reasonably certain to exercise that option
- Periods during which the contract is cancellable if the reporting entity is reasonably certain not to exercise its option to cancel

- Periods covered by an option to extend (or not to terminate) the service in which exercise of the option is controlled by the vendor

The amortization period should be periodically reassessed to determine if it continues to be reasonable. If the estimated amortization period changes (based on new or more recent information), that change should be accounted for prospectively in accordance with ASC 250-10-45-17.

When reassessing the term of the hosting arrangement, a reporting entity should consider the effects of the factors described in SW 3.8, which include obsolescence, competition, other economic factors, and any rapid changes that may be occurring in the development of the arrangement. Additionally, a reporting entity should also consider the effect of any significant implementation costs that are expected to have significant economic value to the reporting entity when the option to extend or terminate the CCA becomes exercisable.

The commencement date for amortization of capitalized CCA implementation costs is determined separately for each module or component; therefore, amortization will begin when a module or component of the CCA is ready for its intended use. This might not be concurrent with the commencement of the CCA service, unless the functionality of a module or component is entirely dependent on the completion of other components. Judgment may be required to allocate implementation costs to multiple modules or components of a CCA.

Although the guidance refers to the "amortization" of capitalized implementation costs, the expense should not be included with other long-lived asset amortization and depreciation. Instead, this expense should be presented in the same line item as the CCA service. For more information on the presentation and disclosure of CCA costs, see FSP 8.6.

Example SW 4-1 illustrates the determination of the amortization period for capitalized CCA implementation costs.

EXAMPLE SW 4-1

Determination of amortization period

On January 1, 20X1, Software Corp enters into a CCA with Cloud Corp for a noncancelable term of three years to access Cloud Corp's cloud-based product design software. At the end of the three-year term, Software Corp has an option to extend the service contract for two years. Software Corp determines that the cloud-based service is critical to Software Corp's operations; additionally, Software Corp incurred significant costs to implement the CCA. As a result, Software Corp concludes it is reasonably certain it will exercise its option to extend the CCA with Cloud Corp at the end of the initial noncancelable term.

What amortization period should Software Corp utilize for capitalized implementation costs related to the CCA?

Analysis

Software Corp should amortize the capitalized implementation costs over five years—the initial three-year noncancelable CCA term plus the optional two-year extension based on the fact that Software Corp is reasonably certain that it will renew the arrangement for another two years. The amortization period should be periodically reassessed to determine if it continues to be reasonable.

Example SW 4-2 illustrates the application of the guidance to a CCA with multiple modules.

EXAMPLE SW 4-2

CCA with multiple modules

During 20X1, Accounting Co enters into a noncancelable two-year SaaS agreement with System Co to implement an enterprise resource planning (ERP) system, which has a general ledger module, an accounts payable module, and an accounts receivable module. The general ledger module must be implemented first and can be utilized without the other two modules.

Through November 30, 20X1, Accounting Co incurred $100,000 in capitalizable implementation costs related to the general ledger module. The general ledger module is expected to be ready in time for the year-end close process in December 20X1. Accounting Co expects to incur $200,000 in additional capitalizable implementation costs during 20X2 related to the remaining two modules of the ERP (accounts payable and accounts receivable).

How should Accounting Co account for the implementation costs related to this arrangement?

Analysis

Accounting Co should capitalize $100,000 of implementation costs attributable to the general ledger module in 20X1. Accounting Co should commence amortization of those capitalized costs in December 20X1 when the general ledger is ready for its intended use because the functionality of the general ledger component is not dependent on the remaining two modules. In 20X2, Accounting Co should capitalize $200,000 of implementation costs for the remaining two modules. Amortization of capitalized costs related to these modules will commence once they are ready for their intended use and be spread over the then-remaining term (i.e., all modules will have a consistent end date).

4.4 *Impairment of capitalized CCA implementation costs*

Capitalized implementation costs for a CCA should be assessed for impairment in accordance with the guidance in ASC 360-10-35 on impairment of long-lived assets—that is, at the asset group level. Further, impairment testing is performed when a triggering event has been identified for the asset group or for an individual asset included in the broader asset group (if significant). ASC 360-10-35-21 provides examples of when to test long-lived assets for recoverability and impairment. For further discussion regarding long-lived asset impairment triggers, see PPE 5.2.3. ASC 350-40-35-11 includes examples of triggering events for capitalized implementation costs for a CCA:

- The hosted CCA service is not expected to provide substantive service potential.
- A significant change occurs in the extent or manner in which the hosting arrangement is used or expected to be used.
- A significant change is made or will be made to the hosting arrangement, including early termination.

When a reporting entity identifies an impairment indicator for capitalized implementation costs for a CCA included in a broader asset group, the reporting entity should consider the significance of the asset in relation to the overall asset group and the facts and circumstances surrounding the impairment indicator to determine whether impairment testing at the asset group level may be required. See PPE 5.2.3.1. These impairment indicators are examples and should not be considered the only potential indicators that an asset or an asset group may not be recoverable.

When a CCA is terminated early, the capitalized implementation costs should be treated similar to an asset being disposed of by abandonment. That is, the implementation costs should be expensed when the related cloud-based service ceases to be used.

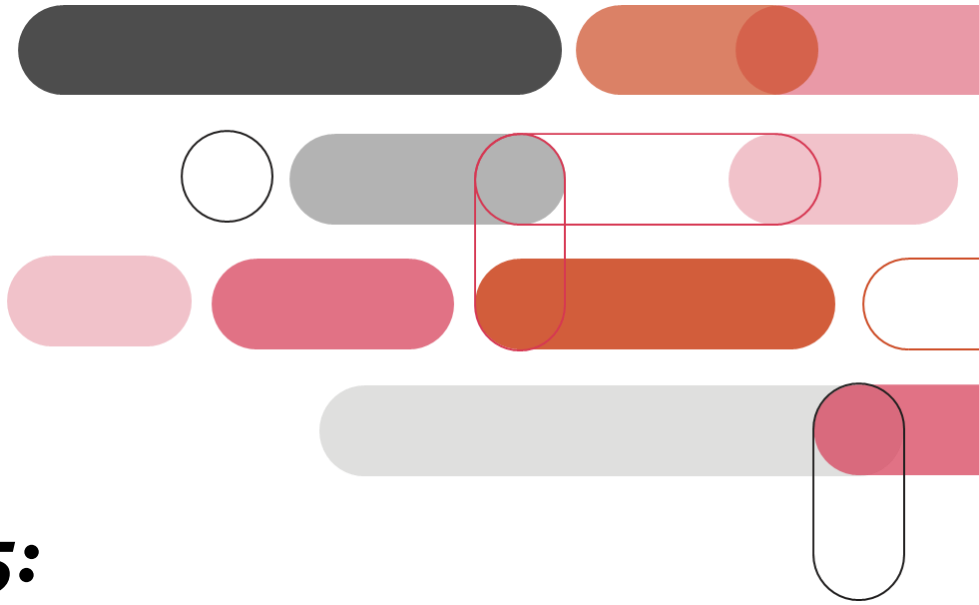
As described in ASC 350-40-35-12, "implementation costs related to each module or component of a hosting arrangement that is a service contract shall be evaluated separately as to when it ceases to be used." Therefore, in the case of a partial termination of the arrangement, the reporting entity should impair the capitalized implementation costs related to the component(s) for which the reporting entity's right to access the hosted software was terminated.

4.5 Transition from a license to a cloud-based solution

A reporting entity may choose to transition their on-premises software to a cloud-based solution. Benefits include reduced costs and decreased resource needs to maintain the infrastructure to house on-premises software. This transition could involve cancelling the on-premises software license or deciding not to renew a license at the end of its term.

If a reporting entity is cancelling a software license or ceasing to utilize the software, the reporting entity should assess the impact to any related unamortized capitalized internal-use software costs. The reporting entity is no longer deriving benefit from the software asset even if the reporting entity is accessing the same software through a CCA service contract. The software asset should be accounted for in accordance with the long-lived assets to be abandoned guidance in ASC 360, *Property, Plant, and Equipment*, as discussed in SW 3.8. The decision to cancel the software license or cease utilizing the software is likely a triggering event that would require an impairment test for the asset group that includes the software asset. Even if the broader asset group does not fail the recoverability test described in ASC 360, adjustment of the useful life of the to-be-abandoned asset may still be necessary in accordance with ASC 360-10-35-47. When the asset ceases to be used, the carrying amount of the asset should equal its salvage value, if any. See SW 3.8 and PPE 6.4.1.

Although a reporting entity may write-off a software asset upon transition to a CCA, costs to implement the new CCA may require capitalization under ASC 350-40, as described in SW 4.2.



Chapter 5: Other software-related costs—updated December 2021

5.1 Other software-related costs - chapter overview—updated April 2025

Reporting entities might incur other software-related costs that are addressed specifically in the accounting guidance, including website development costs (see SW 5.2) and costs of business process reengineering activities (see SW 5.3).

Note about ongoing standard setting

The FASB has an active project related to the accounting and disclosure of software costs. Financial statement preparers and other users of this publication are therefore encouraged to monitor the status of the project, and if finalized, evaluate the effective date of the new guidance and the implications on the accounting and related disclosures.

5.2 Website development costs

The accounting for costs associated with developing and populating an internet website differs in some respects from the guidance on other types of software development. ASC 350-50, *Intangibles—Goodwill and Other—Website Development Costs*, provides guidance on accounting for costs incurred in each of the five stages of website development. The stages of website development are:

- Planning (see SW 5.2.1)
- Website application and infrastructure development (see SW 5.2.2)
- Graphics development (see SW 5.2.3)
- Content development (see SW 5.2.4)
- Operating (see SW 5.2.5)

5.2.1 *Planning stage*

The first stage for website development is the planning stage, which includes activities to plan the website development as described within ASC 350-50-55-2.

ASC 350-50-55-2

Planning stage activities include the following:

a. Develop a business, project plan, or both. This may include identification of specific goals for the website (for example, to provide information, supplant manual processes, conduct e-commerce, and so forth), a competitive analysis, identification of the target audience, creation of time and cost budgets, and estimates of the risks and benefits.

b. Determine the functionalities (for example, order placement, order and shipment tracking, search engine, email, chat rooms, and so forth) of the website.

c. Identify necessary hardware (for example, the server) and web applications. Web applications are the software needed for the website's functionalities. Examples of web applications are search engines, interfaces with inventory or other back-end systems, as well as systems for registration and authentication of users, commerce, content management, usage analysis, and so forth.

d. Determine that the technology necessary to achieve the desired functionalities exists. Factors might include, for example, target audience numbers, user traffic patterns, response time expectations, and security requirements.

e. Explore alternatives for achieving functionalities (for example, internal versus external resources, custom-developed versus licensed software, company-owned versus third-party-hosted applications and servers).

f. Conceptually formulate and/or identify graphics and content (see ASC 350-50-25-8 through 25-13).

g. Invite vendors to demonstrate how their web applications, hardware, or service will help achieve the website's functionalities.

h. Select external vendors or consultants.

i. Identify internal resources for work on the website design and development.

j. Identify software tools and packages required for development purposes.

k. Address legal considerations such as privacy, copyright, trademark, and compliance.

The costs incurred during the planning stage should be expensed as incurred.

5.2.2 *Website application and infrastructure development stage*

The second stage for website development is the application and infrastructure development stage, which includes activities to acquire or develop hardware and software to operate the website.

ASC 350-50-55-3 describes the activities usually performed during this stage.

ASC 350-50-55-3

The website application and infrastructure development stage involves acquiring or developing hardware and software to operate the website. The activities in this stage include the following:

a. Acquire or develop the software tools required for the development work (for example, HTML editor, software to convert existing data to HTML form, graphics software, multimedia software, and so forth).

b. Obtain and register an Internet domain name.

c. Acquire or develop software necessary for general website operations, including server operating system software, Internet server software, web browser software, and Internet protocol software.

d. Develop or acquire and customize code for web applications (for example, catalog software, search engines, order processing systems, sales tax calculation software, payment systems, shipment tracking applications or interfaces, email software, and related security features).

e. Develop or acquire and customize database software and software to integrate distributed applications (for example, corporate databases and accounting systems) into web applications.

f. Develop HTML web pages or develop templates and write code to automatically create HTML pages.

g. Purchase the web and application server(s), Internet connection (bandwidth), routers, staging servers (where preliminary changes to the website are made in a test environment), and production servers (accessible to customers using the website). Alternatively, these services may be provided by a third party via a hosting arrangement.

h. Install developed applications on the web server(s).

i. Create initial hypertext links to other websites or to destinations within the website. Depending on the site, links may be extensive or minimal.

j. Test the website applications (for example, stress testing).

The costs incurred for activities during the website application and infrastructure development stage should be capitalized in accordance with the guidance on internal-use software in ASC 350-40 (see SW 3). This assumes the website is not being developed to be marketed externally (e.g., a vendor sells a website template to a customer to be used in the customer's business).

Costs incurred for website hosting services from a third-party vendor are generally expensed over the period the services are received. If payments are made in advance of receiving the hosting services, prepayments are generally recorded as a prepaid expense. Costs to obtain or register an internet domain name are generally capitalized and amortized over the estimated useful life of the website.

5.2.3 *Graphics development stage*

The third stage for website development is the graphics development stage. ASC 350-50-55-4 and ASC 350-50-55-5 describes the activities generally performed in this stage, which include designing the graphics, layout, and look and feel of the website.

ASC 350-50-55-4

For the purposes of this Subtopic, graphics 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.

ASC 350-50-55-5

Graphics include the design or layout of each page (that is, the graphical user interface), color, images, and the overall look and feel and usability of the website. Creation of graphics may involve coding of software, either directly or through the use of graphic software tools. The amount of coding depends on the complexity of the graphics.

Costs incurred during the graphics development stage for the creation of initial graphics for the website should be capitalized. Subsequent updates to the initial graphics should be expensed as incurred, unless they provide additional functionality. In other words, subsequent updates to the initial graphics should follow the upgrades and enhancements model discussed in SW 3.4.1.

5.2.4 *Content development stage*

The fourth stage for website development is the content development stage. This stage begins after the layout of the website has been designed and includes activities to develop the content that will be displayed. ASC 350-50-55-6 through ASC 350-50-55-8 describes the activities generally performed during this stage.

ASC 350-50-55-6

Content refers to information included on the website, which may be textual or graphical in nature (although the specific graphics described in ASC 350-50-55-4 are excluded from content). For example, articles, product photos, maps, and stock quotes and charts are all forms of content. Content may reside in separate databases that are integrated into (or accessed from) the web page with software, or it may be coded directly into the web pages.

ASC 350-50-55-7

Content may be created or acquired to populate databases or web pages. Content may be acquired from unrelated parties or may be internally developed.

ASC 350-50-55-8

Content is text or graphical information (exclusive of graphics described in ASC 350-50-55-4 through 55-5) on the website which may include information on the entity, products offered, information sources that the user subscribes to, and so forth. Content may originate from databases that must be converted to HTML pages or databases that are linked to HTML pages through integration software. Content also may be coded directly into web pages.

Costs incurred to input content into a website and data conversion costs should be expensed as incurred. Software used to integrate a database with a website should be accounted for in accordance with the internal-use software guidance (see SW 3).

ASC 350-50 does not provide guidance on how to account for the development or the acquisition of the content itself; therefore, other guidance may be applicable, such as ASC 926, *Entertainment—Entertainment Films*, for film content or ASC 928, *Entertainment—Entertainment Music*, for music content. Reporting entities that incur costs to develop or acquire content that is not in the scope of specific guidance, but which is expected to provide economic benefit to the reporting entity as part of the website over multiple future periods should consider whether it is appropriate to apply this guidance by analogy.

5.2.5 *Operating stage*

The last stage for website development is the operating stage, which occurs once the website has been fully designed and is operational. ASC 350-50-55-9 describes the activities usually performed during this stage.

ASC 350-50-55-9

Costs incurred during the operating stage include training, administration, maintenance, and other costs to operate an existing website. Activities in the operating stage include the following:

a. Train employees involved in support of the website.

b. Register the website with Internet search engines.

c. Perform user administration activities.

d. Update site graphics (for updates of graphics related to major enhancements, see [h]).

e. Perform regular backups.

f. Create new links.

g. Verify that links are functioning properly and update existing links (that is, link management or maintenance).

h. Add additional functionalities or features.

i. Perform routine security reviews of the website and, if applicable, of the third-party host.

j. Perform usage analysis.

Generally, the costs incurred during the operating stage (including costs to register the website with internet search engines) should be expensed as incurred. However, if costs incurred during this stage involve providing additional functions or features, those costs should be evaluated similar to software upgrades and enhancements to determine whether they should be capitalized or expensed (see SW 3.4.1). ASC 350-50-25-16 indicates that if reporting entities cannot reasonably separate the costs of relatively minor upgrades and enhancements from costs of maintenance, those costs should be expensed as incurred.

5.3 Costs of business process reengineering activities

ASC 720-45, *Other Expenses—Business and Technology Reengineering*, prescribes that reporting entities expense as incurred the costs of business process reengineering activities, whether incurred internally or by third parties. This guidance also applies when the business process reengineering activities are part of a project that includes plans to acquire, develop, or implement internal-use software or other assets. However, this guidance does not change the accounting for internal-use software development costs or the acquisition of long-lived assets.

ASC 720-45-25-2

The following third-party or internally generated costs typically associated with business process reengineering shall be expensed as incurred:

a. Preparation of request for proposal—the process of preparing a proposal.

b. Current state assessment—the process of documenting the entity's current business process, except as it relates to current software structure. This activity is sometimes called mapping, developing an as-is baseline, flow charting, and determining current business process structure.

c. Process reengineering—the effort to reengineer the entity's business process to increase efficiency and effectiveness. This activity is sometimes called analysis, determining best-in-class, profit and performance improvement development, and developing should-be processes.

d. Restructuring the workforce—the effort to determine what employee makeup is necessary to operate the reengineered business processes.

Costs associated with the acquisition or the construction of property and equipment related to a business process reengineering project should be accounted for in accordance with the reporting entity's existing policies for accounting for productive assets. Refer to PPE 1 for information on the capitalization of costs associated with long-lived assets. Similarly, costs related to the development or acquisition of internal-use software in connection with a business process reengineering project should be accounted for using the framework for internal-use software costs (see SW 3).

Refer to ASC 720-45 for additional examples of costs incurred for business process reengineering and information technology transformation projects and the related accounting conclusions.