

Chapter 4 — Software and Software-Related Costs

4.1 Background

As technology evolves, entities typically incur myriad costs related to software. For example, cloud-based arrangements have revolutionized the business and technology landscape, offering more flexible and often lower-cost IT solutions that allow businesses to outsource their traditional enterprise resource planning (ERP) systems or any other on-site application to an off-site, on-demand solution. In addition, an increasing number of processes are managed by using automated solutions, such as CRM, human resources, payroll, finance, and collaboration and communication tools. This has resulted in entities' incurring increasing amounts of software-related costs as they either purchase licenses to on-premise software products or contract with vendors to access and use software solutions over the Internet (e.g., cloud computing or SaaS). Entities also frequently use hybrid deployments, in which they purchase or develop on-premise software (some of which may be deployed in a private cloud environment) and use that software in conjunction with another cloud-based third-party platform (i.e., a public cloud). Further, entities may incur costs to develop software for their own internal use as well as for external sales to customers. Entities incurring such costs will need to determine whether they represent assets that can be capitalized under the applicable accounting standards. Different accounting guidance exists for costs related to software that is (1) sold, leased, or marketed; (2) obtained or developed for internal use; and (3) accessed in a cloud-based (or hosting) arrangement that is a service contract.

It is important to determine whether software costs incurred are within the scope of ASC 985-20 or ASC 350-40 because the requirements for capitalization vary significantly between the two standards. For example, ASC 985-20-25-1 states, in part, that "[a]ll costs incurred to establish the technological feasibility of a computer software product to be sold, leased, or otherwise marketed are research and development costs." Once technological feasibility is established, the costs of producing product masters, including coding and testing, are generally capitalized until the product is available for general use.¹ Because technological feasibility is often established shortly before the software product reaches the general availability stage, many software entities do not have material costs capitalized under ASC 985-20.

By contrast, ASC 350-40 does not require the establishment of technological feasibility for capitalization but does have other requirements for capitalization depending on the stage of development. Generally, development costs incurred during the application development stage are capitalized, while costs incurred during the preliminary-project and post-implementation-operation stages are expensed as incurred. Costs incurred for internal-use software will typically meet the capitalization requirements earlier in the development cycle than costs incurred for software licensed externally. As a result, more costs typically qualify for capitalization when software is obtained or developed for internal use than those for software that is licensed externally. Further, ASC 350-40 also applies to implementation

¹ Production costs for software that is to be used as an integral part of a product or process cannot be capitalized until both (1) technological feasibility has been established and (2) all R&D activities for the other components of the product or process have been completed.

costs incurred for cloud-based (or hosting) arrangements that are service contracts. Generally, implementation costs incurred for such arrangements during the application development stage are deferred and presented in prepaid expenses, while other costs (e.g., cloud computing and hosting costs) are expensed as incurred (unless they are related to other capitalizable assets such as hardware and on-premise software). Within ASC 350-40, guidance differs for cloud-based (or hosting) arrangements versus internal-use software (e.g., only implementation costs for cloud-based [or hosting] arrangements are eligible for deferral, and there are different presentation requirements).

Because of the above differences in capitalization requirements, the application of the incorrect guidance could have material accounting implications. In addition, complexities may arise when entities evaluate the appropriate scope as technology evolves and business models shift. For example, entities may transition from using software solutions internally to selling and marketing them. Similarly, entities may shift their business model from selling on-premise licensed solutions to SaaS arrangements. It is therefore important to understand the scope guidance and regularly reassess previous scope conclusions in a dynamic environment.

4.2 Scope Considerations

4.2.1 On-Premise Licensed Software

ASC 985-20-15-2 states that ASC 985-20 applies to the costs of “computer software to be sold, leased, or otherwise marketed as a separate product or as part of a product or process, whether internally developed and produced or purchased.” Typically, software within the scope of ASC 985-20 is licensed on a nonexclusive on-premise basis, either as a perpetual or term-based (i.e., subscription-based) license, with the sale of such software accounted for under ASC 606.

In assessing how software development costs should be accounted for, entities must determine whether there is a substantive plan to market the software externally or whether one will be created during the software’s development period. If either is the case, the software development costs will be subject to ASC 985-20 (see [Section 4.2.2.2](#)).



Connecting the Dots

Some on-premise software applications, such as mobile applications (apps), may not be licensed for consideration. In those circumstances, an entity must carefully evaluate whether the software is considered “sold, leased, or otherwise marketed as a separate product or as part of a product or process” under ASC 985-20. For example, an entity may sell gaming apps for consideration, and such apps would therefore be within the scope of ASC 985-20. However, gaming apps may also be offered on a “freemium” basis, with in-app sales (e.g., consideration paid to play a game without viewing ads or for virtual items that enhance the gaming experience). Even though a gaming app itself is free for download, we believe that it would still be considered “marketed as a separate product,” particularly since there could be in-app sales for consideration. Further, apps may be sold as part of a product or process (see [Section 4.2.1.2](#)). For example, a thin-client app may be sold as part of a cloud-based service, but if its sole function is to enable connection to the cloud-based service, it may not be substantive enough to be considered “sold, leased, or otherwise marketed.” Therefore, an entity may need to use judgment to determine whether apps that are free for download or part of a product or process are within the scope of ASC 985-20. If they are not within the scope of ASC 985-20, they could be within the scope of ASC 350-40 (see [Section 4.2.2](#)).

4.2.1.1 *Software Product*

A software product is defined by ASC 985-20-55-1 as having the following qualities:

- “As a product, it is complete and has exchange value.”
- “As software, it is a set of programs that interact with each other. A program is further defined as a series of instructions or statements that cause a computer to do work.”

A software product is a set of programs (e.g., software code) that has been packaged in such a way that it can be marketed to third parties. The software product may be sold to either end users or distributors. A software product also consists of the appropriate documentation and training materials. Determining whether a set of programs consists of a single software product or multiple software products requires judgment since ASC 985-20 does not provide specific guidance on the unit of accounting.

When determining separate software products, an entity should consider how programs are marketed. A set of programs that is separately priced and marketed would most likely be treated as a separate software product. For example, programs may be packaged and priced differently depending on the market (e.g., different geographic areas or industries). In that circumstance, each set of packaged programs may be a separate software product, with costs identified and allocated through the use of a reasonable method.

An entity could also consider the functionality and interdependence of its programs. For example, two sets of technically independent programs, for which costs can be separately identified and a basis for allocating revenue can be established, may be two separate software products. A set of programs is technically independent if other programs are not essential to the set's functionality. Therefore, the entity might be able to market that set as a separate software product because customers will be able to effectively use it without any other programs. By contrast, sets of programs that are technically interdependent may not be marketed separately. For example, if a set of programs has been developed but has no stand-alone functionality without the development of additional programs, it most likely would not be a separate software product.

A newly developed set of programs could be combined with an existing separate software product if it is integrated with and intended to replace that product. In addition, modules or add-ons with different features and functions can be developed for an existing separate software product. If a set of programs associated with a module or add-on is separately priced, it may be treated as its own separate software product. However, if that set of programs is not priced separately and revenue cannot be reasonably allocated to it, it should be treated as part of the existing software product.

A software product can either be developed by an entity's own employees or by third parties. A developer also can acquire an existing software product from a third party. Because there is no specific ownership requirement in ASC 985-20, an entity may obtain the marketing rights to licensed software (e.g., as a reseller or distributor), and the amount paid to obtain those rights would be a cost of a separate software product (or part of another software product) as though the entity had acquired or developed the program itself (i.e., as though it owned the IP outright).

4.2.1.2 *Software That Is Part of a Product or Process*

While software often is sold as a product that has stand-alone functionality (e.g., software used to process tax returns), software may also be embedded as part of another product that is sold, such as firmware that is embedded in smart devices (e.g., smartphones, tablets, gaming consoles, and other devices associated with the IoT).

Further, software could be sold as part of a process. While not specifically defined in ASC 985-20, a process is described in ASC 730-10-15-3 as “a system whose output is to be sold, leased, or otherwise marketed to others. A process also may be used internally as a part of a manufacturing activity or a service activity where the service itself is marketed.” Therefore, if on-premise software is sold as part of a service, it would be subject to ASC 985-20. For example, an entity could sell a customer on-premise payroll software that enables the entity to provide payroll and tax services to that customer (i.e., the customer uses the on-premise software in connection with the payroll and tax services it receives from the entity).



Connecting the Dots

Determining whether software is sold as part of a product or process could require judgment. If software is used in the design, development, or manufacturing of a separate product or service, the software would not be within the scope of ASC 985-20 unless that software is included in the product or service sold. For example, if software is used to produce an architectural blueprint but only the output associated with the blueprint is sold to a customer, that software would not be within the scope of ASC 985-20. On the other hand, if the software is also provided with the architectural blueprint sold to the customer so that the customer can modify the architectural blueprint, that software would be within the scope of ASC 985-20.

4.2.1.3 Software Sold as Part of a Hosting Arrangement

Sometimes, software may be sold as part of a hosting arrangement,² such as SaaS that is accessed via an online portal. If so, the software is subject to ASC 985-20 only if both of the following criteria in ASC 985-20-15-5 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.



Connecting the Dots

Some may question whether “at any time” during the hosting period means *at every point of time* during the hosting period. We do not believe that to be the case. For example, an entity’s arrangements may specify that the customer will automatically obtain the software at the end of the hosting period. We believe that as long as the customer can take possession of the software at that point without significant penalty and it is feasible for the customer to run the software (either on its own or with a third-party vendor), the software license is a separate promise in the hosting arrangement and would therefore meet the criteria in ASC 985-20-15-5(a) and (b).

If the above criteria are met, an entity (i.e., the vendor) would account for only the software costs under ASC 985-20. It would account for costs associated with hosting the software separately under other U.S. GAAP. For example, if the entity purchases servers to provide the hosting service, it would account for those servers as long-lived assets under ASC 360.

ASC 985-20-15-6 states that the term “significant penalty” as used in ASC 985-20-15-5(a) contains the following two distinct concepts:

- a. The ability to take delivery of the software without incurring significant cost
- b. The ability to use the software separately without a significant diminution in utility or value.

² A hosting arrangement is defined in the ASC master glossary as being “[i]n connection with accessing and using software products, an arrangement in which the customer of the software does not currently have possession of the software; rather, the customer accesses and uses the software on an as-needed basis.”

The analysis for determining whether either or both of these conditions are met (i.e., a significant penalty exists), depends on the facts and circumstances of the arrangement and requires judgment. An entity may consider the following factors (not all-inclusive) in making this assessment:

- Contractual cancellation fees associated with the hosting arrangement.
- Other contractual penalties for taking possession of the software (e.g., the requirement that the customer continue to pay the hosting fees for the remainder of the hosting term even though hosting services are terminated).
- Costs of transitioning to (1) use of the software on the customer's own servers or (2) hosting of the software by the customer's third-party vendor.
- Whether the utility and value of the software can be maintained upon transition (e.g., whether (1) the customer will continue to receive updates, upgrades, and enhancements and (2) the software will be capable of providing the same functionality in another environment).
- Whether the software (1) has stand-alone functionality (on its own or with readily available resources) or (2) is significantly tied to other products or services that can be provided only by the entity and will no longer be provided if the customer takes possession of the software.

Significance can be evaluated both quantitatively and qualitatively. The accounting literature does not contain specific guidance on (1) which elements of the contract should be included in the measurement of the amount of the penalty or (2) the benchmark against which the entity should measure the amount of the penalty when determining whether the penalty is quantitatively significant. An entity may have an established policy for determining whether the penalty is significant. For example, in a manner consistent with other Codification subtopics, the entity may reasonably conclude that amounts above 10 percent of a given benchmark are significant. Establishing a method of determining both the elements of the contract to include in the measurement of the penalty and the benchmark against which to measure the penalty is an accounting policy decision that the entity should apply consistently. See [Section 2.4.4.1](#) for illustrative examples of how an entity might evaluate the existence of a significant penalty.

If the criteria in ASC 985-20-15-5 are not met (i.e., the customer does not receive on-premise software), the entity accounts for the software costs under ASC 350-40 as internal-use software. However, the entity must evaluate all its arrangements. If it has other substantive arrangements in which the same software is sold, leased, or marketed (i.e., sold as on-premise software), the entity must account for the software costs under ASC 985-20 (see [Section 4.2.3](#)).

4.2.2 Internal-Use Software

ASC 350-40-15-2A describes internal-use software as having 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.

ASC 350-40-55-1 and 55-2 contain the following examples of fact patterns in which software is for internal use or not for internal use:

ASC 350-40

55-1 The following is a list of examples illustrating when computer software is for internal use:

- a. A manufacturing entity purchases robots and customizes the software that the robots use to function. The robots are used in a manufacturing process that results in finished goods.
- b. An entity develops software that helps it improve its cash management, which may allow the entity to earn more revenue.
- c. An entity purchases or develops software to process payroll, accounts payable, and accounts receivable.
- d. An entity purchases software related to the installation of an online system used to keep membership data.
- e. A travel agency purchases a software system to price vacation packages and obtain airfares.
- f. A bank develops software that allows a customer to withdraw cash, inquire about balances, make loan payments, and execute wire transfers.
- g. A mortgage loan servicing entity develops or purchases computer software to enhance the speed of services provided to customers.
- h. A telecommunications entity develops software to run its switches that are necessary for various telephone services such as voice mail and call forwarding.
- i. An entity is in the process of developing an accounts receivable system. The software specifications meet the entity's internal needs and the entity did not have a marketing plan before or during the development of the software. In addition, the entity has not sold any of its internal-use software in the past. Two years after completion of the project, the entity decided to market the product to recoup some or all of its costs.
- j. A broker-dealer entity develops a software database and charges for financial information distributed through the database.
- k. An entity develops software to be used to create components of music videos (for example, the software used to blend and change the faces of models in music videos). The entity then sells the final music videos, which do not contain the software, to another entity.
- l. An entity purchases software to computerize a manual catalog and then sells the manual catalog to the public.
- m. A law firm develops an intranet research tool that allows firm members to locate and search the firm's databases for information relevant to their cases. The system provides users with the ability to print cases, search for related topics, and annotate their personal copies of the database.

ASC 350-40 (continued)

55-2 The following list provides examples of computer software that is not for internal use:

- a. An entity sells software required to operate its products, such as robots, electronic game systems, video cassette recorders, automobiles, voice-mail systems, satellites, and cash registers.
- b. A pharmaceutical entity buys machines and writes all of the software that allows the machines to function. The pharmaceutical entity then sells the machines, which help control the dispensation of medication to patients and help control inventory, to hospitals.
- c. A semiconductor entity develops software embedded in a microcomputer chip used in automobile electronic systems.
- d. An entity purchases software to computerize a manual catalog and then sells the computer version and the related software to the public.
- e. A software entity develops an operating system for sale and for internal use. Though the specifications of the software meet the entity's internal needs, the entity had a marketing plan before the project was complete. In addition, the entity has a history of selling software that it also uses internally and the plan has a reasonable possibility of being implemented.
- f. An entity is developing software for a point-of-sale system. The system is for internal use; however, a marketing plan is being developed concurrently with the software development. The plan has a reasonable possibility of being implemented.
- g. A telecommunications entity purchases computer software to be used in research and development activities.
- h. An entity incurs costs to develop computer software for another entity under a contract with that other entity.

In many cases, it will be obvious that software is obtained or developed solely to meet an entity's internal needs (e.g., ERP software purchased from a third-party vendor and used solely by the entity to process business transactions). In other circumstances, entities will need to carefully evaluate the manner in which the software is or will be used to determine whether it is subject to ASC 350-40.

In addition, the guidance in ASC 350-40 must be applied at the individual component or module level. While there is no specific guidance on what an individual component or module might be, an entity could consider the level of functionality each component or module provides as well as the level of interdependence between the components or modules.



Connecting the Dots

ASC 350-40-15-2 provides an example of an accounting software system that contains separate components or modules, including a general ledger, an accounts payable subledger, and an accounts receivable subledger. Determining the appropriate components or modules is important because the assessment of amortization and impairment for abandonments is performed at the component or module level. In addition, components or modules of a particular software system may be at different stages of development, and costs would need to be separately tracked, particularly in agile development environments.

4.2.2.1 *Software Is Purchased for Internal Use as Part of a Hosting Arrangement*

An entity may obtain internal-use software as part of a hosting arrangement with a vendor. If so, the software costs are subject to ASC 350-40 if both of the following criteria in ASC 350-40-15-4A 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 criteria above are the same as those in ASC 985-20-15-5 (see [Section 4.2.1.3](#)). If the criteria are met, the costs associated with the purchase or license of the software are subject to ASC 350-40. If the criteria are not met, the arrangement is a service contract (see [Section 4.2.5](#)).



Connecting the Dots

It is common for software to be hosted on a third-party platform or infrastructure (i.e., not the vendor's or customer's on-site platform or infrastructure). In these circumstances, it is important to determine who has the contract with that third party (i.e., whether it is the vendor's or customer's cloud instance of the third-party platform or infrastructure). If the software is hosted on an entity's (i.e., a customer's) cloud instance and the entity has possession of the software, the costs associated with it are subject to ASC 350-40. By contrast, if the software is hosted on the vendor's cloud instance and the entity (i.e., the customer) cannot otherwise obtain possession of the software without significant penalty, the costs associated with that software are accounted for as a service arrangement and only the implementation costs are subject to ASC 350-40.

4.2.2.2 *No Substantive Plan to Market the Software Externally*

If the software is or will be marketed externally (i.e., marketed to be sold or licensed on an on-premise basis), the costs will be within the scope of ASC 985-20. Therefore, if a substantive plan to market the software externally exists or is being developed during the software development period, regardless of whether the software is also intended to meet an internal need, the costs will be subject to ASC 985-20. The software must be intended solely for internal use to be subject to ASC 350-40.

To be considered "substantive," a marketing plan needs to be sufficiently detailed, and its implementation should be reasonably possible.³ ASC 350-40-15-2B states that a substantive plan "could include the selection of a marketing channel or channels with identified promotional, delivery, billing, and support activities." It also states that "routine market feasibility studies are not substantive plans to market software."



Connecting the Dots

When an entity is determining whether it has a substantive plan to market software externally, it must under ASC 350-40 evaluate its past practices and patterns. For example, if the entity has a past practice or pattern of both using software internally and selling that same software externally (or deciding to market internal-use software externally during development), a rebuttable presumption is created that any software developed by the entity is intended for sale, lease, or marketing (i.e., the software costs are subject to ASC 985-20).⁴

Example 4-1

Company A, a recording and music distribution company, is developing software that would enable users to listen to, edit, and record music files. Company A plans to use the software to create music albums that it will then sell to customers. Company A is also negotiating with four software resellers to sell them the new product. Company A's marketing department is compiling a detailed plan and designing promotional material for the new product, and implementation of the marketing plan is considered at least reasonably possible. Therefore, A has a substantive marketing plan and should account for the costs of the new software product under ASC 985-20.

³ The ASC master glossary defines reasonably possible as "[t]he chance of the future event or events occurring is more than remote but less than likely."

⁴ See ASC 350-40-15-2C and ASC 350-40-35-10.

Example 4-2

Company B is developing software that would enable it to better manage its advertising campaigns. Company B has engaged a market research firm to conduct a market survey to determine whether a market for the new software product exists. The market survey conducted by the market research firm is a routine market feasibility study and not a substantive plan to market the product. Therefore, unless and until there is a substantive plan being developed to market the software to others, B should account for the costs of the software product under ASC 350-40.

Example 4-3

Company C is developing a data management software platform that will be sold only as a cloud-based arrangement (i.e., as internal-use software; see [Section 4.2.2.3](#)). It does not have a marketing plan or intent to sell the software on an on-premise basis (i.e., customers will not have the contractual ability to take possession of the software). However, C has a past practice of selling other software products to customers on both a hosted basis and on an on-premise basis, depending on the customer's request. Therefore, while C has neither a marketing plan nor the intent to sell the data management software on an on-premise basis, its past practice creates a rebuttable presumption that the data management software is intended for sale, lease, or marketing.

Company C considers that recently, it has either transitioned or is in the process of transitioning its customers to using all of its software products on a hosted basis. In addition, for any new or modified arrangements, customers will no longer have the contractual ability to take possession of any of C's software products. Therefore, C concludes that it can overcome the rebuttable presumption that the data management software is intended for sale, lease, or marketing and the data management software is therefore subject to ASC 350-40.

4.2.2.3 Software Is Marketed or Sold Only as a Cloud-Based (or Hosting) Arrangement

If software is being marketed or sold only as a cloud-based (or hosting) arrangement, that software would be considered internal-use software. To determine whether the product is considered software to be sold, leased, or marketed, and therefore accounted for under ASC 985-20, see [Section 4.2.1](#).

Many cloud-based or hosting arrangements include a “license” to software but allow the customer to use the software only in an entity's (rather than the customer's) hosted environment (because of contractual or practical limitations, or both, to taking possession of the software). The entity's hosted environment could include its cloud instance on a third party's platform or infrastructure (i.e., the entity has a contract with a third party to host its software). Although these arrangements may include a contractual license, if the customer is unable to take possession of the software subject to the license without significant penalty, the software is for the entity's internal use in providing a service to its customers. These transactions are accounted for as service arrangements (rather than licensing arrangements) since the entity is providing the functionality of the software through a hosting arrangement (service) rather than through an actual on-premise software license that is controlled by the customer. Therefore, the costs to develop or acquire such software should be accounted for under ASC 350-40.



Connecting the Dots

ASC 350-40-15-5 specifies that software is for internal use (vs. sold as on-premise software) if it is used to develop a product or provide a service sold to a customer but the customer does not actually acquire the software or a future right to use it.

Example 4-4

Company D offers its office productivity software solution as a SaaS whereby its customers access the solution through an online portal and store data on D's secure servers. The software will always be maintained at the most up-to-date version available, and customers have rights to online and telephone support. Customers do not have the ability to take possession of the software.

Because customers are not permitted to take possession of the software and may use only D's cloud-based service, D concludes that the costs associated with its office productivity software should be accounted for under ASC 350-40.

Example 4-5

Company E is developing a CRM software solution to be marketed and sold to customers. Company E also intends to use the software internally to manage its communications and relationships with customers and potential customers.

A detailed marketing plan has already been developed for the software. The software will be provided to customers on a hosted basis (i.e., the software will be accessed by using an Internet connection) and will connect to E's proprietary data analytics platform, which has already been developed and is housed on E's own servers (i.e., it is a SaaS solution that is accessed only online). Company E's data analytics platform will be a significant part of the overall solution sold to its customers and will be significantly integrated with the CRM software solution being developed. Company E plans to provide its customers with the contractual ability to take possession of the CRM software on an on-premise basis, when requested at any point during the hosting period, without paying E a penalty or cancellation fee. However, customers will not have the contractual ability to take possession of E's data analytics platform. In addition, cancellation of the hosting service for the CRM software will also result in the cancellation of the SaaS for E's data analytics platform, which cannot be easily replicated by the customer or third-party vendors. Further, customers would incur significant costs to integrate the CRM software with other third-party data analytics platforms.

While a customer will have the "contractual right to take possession of the software at any time during the hosting period" without paying E a penalty or cancellation fee, it cannot do so without incurring a significant penalty (i.e., significant diminution in utility or value of the CRM software without E's data analytics platform). Therefore, E concludes that the software costs incurred to develop the CRM software should be accounted for under ASC 350-40.

4.2.3 Transition Between Internal-Use Software and On-Premise Licensed Software

4.2.3.1 Transition to Licensing Software Externally

After the development of internal-use software, an entity may decide to license the software externally on an on-premise basis. If so, the entity must first account for any proceeds received from the license of the software, net of any direct incremental costs (e.g., commissions, software reproduction costs, warranty and service obligations, and installation costs), as a reduction of the carrying amount of any costs for that software that were capitalized under ASC 350-40. It cannot recognize profit on the software until it has reduced the carrying amount to zero. When the entity has reduced the carrying amount to zero (inclusive of any amortization of the software), it can then recognize subsequent proceeds as revenue under ASC 606 (or a gain under ASC 610-20 if the contract is not with a customer).⁵ Any subsequent software development costs for that software product are then subject to ASC 985-20.

⁵ See ASC 350-40-35-7 and 35-8.

If the decision to market the software externally happens during its development, any software costs incurred prospectively are accounted for under ASC 985-20. As indicated above, this decision should be supported by a substantive plan before the entity switches to ASC 985-20. In addition, amortization and impairment assessments should likewise be subject to ASC 985-20.⁶

4.2.3.2 Transition to Providing Software Through a Cloud-Based Arrangement

Because there have been significant shifts over time to migrate software solutions to the cloud, it is common for software entities to sell software on both an on-premise licensed basis and a cloud basis. In those circumstances, any software costs are subject to ASC 985-20.

However, scope-related questions have arisen in situations in which an entity predominantly sells and provides a software solution through cloud-based arrangements. As long as there continue to be substantive external sales of on-premise software, we believe that the software costs should still be subject to ASC 985-20. If, instead, an entity no longer has substantive external sales of on-premise software, neither ASC 985-20 nor ASC 350-40 provides transition guidance. In that circumstance, we believe that it is reasonable to account for any future software development costs in accordance with ASC 350-40 and to account for the aggregate amount of capitalized software costs for the software prospectively under ASC 350-40 (e.g., amortization and impairment). We believe that an entity may apply judgment in determining whether there are any substantive external sales of on-premise software.

4.2.4 Hybrid Cloud-Based Software Solutions

Many entities sell hybrid cloud-based software solutions, in which on-premise licensed software is sold with cloud-based software. Often, the on-premise licensed software interacts with the cloud-based software, and in some circumstances, the on-premise licensed software may be significantly integrated, interdependent, or interrelated with the cloud-based software.

In these situations, an entity must carefully track its software costs to determine which are (1) subject to ASC 985-20 (because there are substantive sales of on-premise licensed software) or (2) subject to ASC 350-40 (because the software is sold only as a service). Even if the on-premise software is significantly integrated, interdependent, or interrelated with the cloud-based software, it generally would not be appropriate to account for all software costs under ASC 985-20 if the software that is sold only as a service is substantive. Likewise, it generally would not be appropriate to account for all software costs under ASC 350-40 if the software sold on an on-premise licensed basis is substantive.

Example 4-6

Company F has a database management system, which is software that it uses in delivering its information services to customers through an online portal. The system collects data from real-time feeds, news sources, and contributed data sources. Company F also sells an on-premise license to its data analytics and machine learning software product, which includes an interface to F's database management system and is downloaded on a customer's servers.

The costs incurred in connection with the database management system are within the scope of ASC 350-40. However, the costs of the data analytics and machine learning software product, which resides on a customer's servers, are accounted for under ASC 985-20. Therefore, F should separately track its software costs for each software solution.

⁶ See ASC 350-40-35-9.

4.2.5 Cloud-Based (or Hosting) Service Arrangements

An entity may enter into a cloud-based (or hosting) arrangement with a vendor (typically for internal use). In determining whether it has purchased a software license or a service arrangement, the entity must evaluate the same considerations as described in [Section 4.2.2.1](#). If the entity (1) does not have “the contractual right to take possession of the software at any time during the hosting period without significant penalty” or (2) it is not “feasible for the [entity] to either run the software on its own hardware or contract with another party unrelated to the vendor to host the software,” the entity has entered into a service contract.⁷ In this circumstance, only implementation costs incurred would be subject to ASC 350-40. An entity may need to use judgment in determining which costs are related to implementation — “implementation cost” is not a defined term because, as paragraph BC14 of [ASU 2018-15](#) states, “[ASC] 350-40 already has appropriate guidance that entities currently apply in practice.” While the term is not defined in ASC 350-40, the EITF and members of the FASB board and staff discussed the following examples of implementation activities during the October 12, 2017, EITF meeting:⁸

- “Integration (developing interfaces between the hosted software and the entity’s other systems).”
- “Customization, either of the entity’s other systems or of the hosted software.”
- “Configuration, either of the entity’s other systems or of the hosted software.”
- “Data conversion or migration.”
- “Installation.”
- “Architecture and design.”
- “Coding.”
- “Testing.”
- “Training.”
- “Business process reengineering.”



Connecting the Dots

When an entity incurs implementation costs for a cloud-based (or hosting) service arrangement, it may also purchase or develop internal-use software as part of that implementation. In that circumstance, the entity should separately account for the costs incurred for that internal-use software under ASC 350-40.

In accordance with ASC 350-40, capitalized implementation costs related to cloud-based service arrangements are deferred and presented in the statement of financial position as prepaid expenses. Further, when such costs are amortized, they are (1) presented in the income statement as an operating expense with the associated hosting fees and (2) classified in the statement of cash flows as an operating activity.

⁷ See ASC 350-40-15-4A through 15-4C.

⁸ Quoted text is from [Issue Summary No. 1, Supplement No. 1](#), which the FASB staff released on September 28, 2017, to facilitate discussion of EITF Issue 17-A at the October 12, 2017, EITF meeting.

4.2.6 Multiple-Element Arrangements

Entities that purchase internal-use software or cloud-based services often purchase multiple elements in the same arrangement (e.g., on-premise software licenses, PCS, cloud-based services, and professional services). ASC 350-40-30-4 requires entities to allocate the cost to all individual elements on the basis of their stand-alone prices.⁹

Example 4-7

Company G purchases a three-year noncancelable software subscription from Vendor H that enables G to manage its data center (e.g., manage various IT workloads). The software can operate (1) on different types of commodity hardware that G can purchase and use on its own premises or (2) through cloud computing arrangements (both the hardware and cloud computing services can be purchased from various third-party vendors and are not part of the arrangement between G and H). The subscription arrangement includes a three-year term-based license that is delivered digitally (it can be downloaded on G's own servers or third-party servers if G chooses to access it through its vendor's cloud computing platform), as well as support and maintenance over the three-year term. Company G also purchases professional services, including training and business process reengineering services, from H in the same subscription arrangement. Company G determines that there are three elements in the arrangement and allocates the total consideration payable to H to those elements on the basis of their relative stand-alone prices. The three elements are accounted for as follows:

- Because G takes possession of the on-premise term-based software license, the amount allocated to it is capitalized as internal-use software under ASC 350-40. The capitalized software cost is then amortized on a straight-line basis over the three-year term and is subject to the impairment guidance in ASC 360.
- The amount allocated to the support and maintenance is expensed over the three-year term. If G prepays for the support and maintenance, it is initially recognized as a prepaid expense.
- The amount allocated to the professional services is expensed as incurred. If G prepays for the professional services, it is initially recognized as a prepaid expense.

4.2.7 Other Guidance to Consider

Software-related costs may be subject to U.S. GAAP other than ASC 985-20 or ASC 350-40. The discussion below describes other guidance that may apply to such costs.

4.2.7.1 Web Site Development Costs

Web site development costs are subject to ASC 350-50. The guidance is similar to that in ASC 350-40. For example, under ASC 350-50-25-6, if software for a Web site is purchased or developed for an entity's internal needs, costs incurred for (1) purchased software tools or (2) internally developed software tools during the application development stage are generally capitalized. In addition, certain software acquired or developed for internal use related to Web site operation or graphics is directly within the scope of ASC 350-40.

While ASC 350-50 refers to Web site content, it does not address the accounting for such content. Therefore, Web site content is accounted for under other U.S. GAAP. For example, if an entity is a licensee in the record and music industry and relicenses music content, it would apply the guidance in ASC 928-340.

⁹ A stand-alone price is defined in ASC 350-40-20 as the "price at which a customer would purchase a component of a contract separately."

4.2.7.2 *Software Used for Research and Development Activities*

If software is used in R&D activities and does not have alternative future uses, it is subject to ASC 730-10. In addition, the following software costs are accounted for as R&D costs:

- For software subject to ASC 985-20, all costs incurred before the establishment of technological feasibility.¹⁰
- For software subject to ASC 350-40, all costs for pilot projects (i.e., “[d]esign, construction, and operation of a pilot [project] that is not of a scale economically feasible to the entity for commercial production”).¹¹
- For software subject to ASC 350-40, all costs associated with a particular R&D project, “regardless of whether the software has alternative future uses.”¹²

Example 4-8

Company J is trying to implement a supply management system by using blockchain technology but is not certain that the supply management system can be designed to meet J's internal requirements. Company J has decided to implement its system at one of its smaller facilities for 90 days to determine whether the system will function as intended.

A project of this nature would be considered a pilot project and accounted for as R&D because J is implementing a software system, intended for company-wide implementation, on a small scale. Often these pilot projects are implemented at locations at which the risk of loss is very low or the cost to run parallel systems is not significant.

Software associated with R&D assets may be acquired in a business combination. If the software will be used for R&D activities, they are subject to the guidance in ASC 805-20 and ASC 350-30. In accordance with ASC 805-20, they are recognized as an asset and measured at fair value.

4.2.7.3 *Significant Production, Modification, or Customization of Software*

Software sold to customers in arrangements that require significant production, modification, or customization is accounted for under ASC 606. If the software is being produced, modified, or customized for a specific customer contract, the costs for such software represent fulfillment costs that are subject to ASC 340-40.

4.2.7.4 *Business Process Reengineering Activities*

An entity may incur costs associated with business process reengineering activities as part of developing software or implementing cloud-based solutions. Those costs are subject to ASC 720-45 and are expensed as incurred.

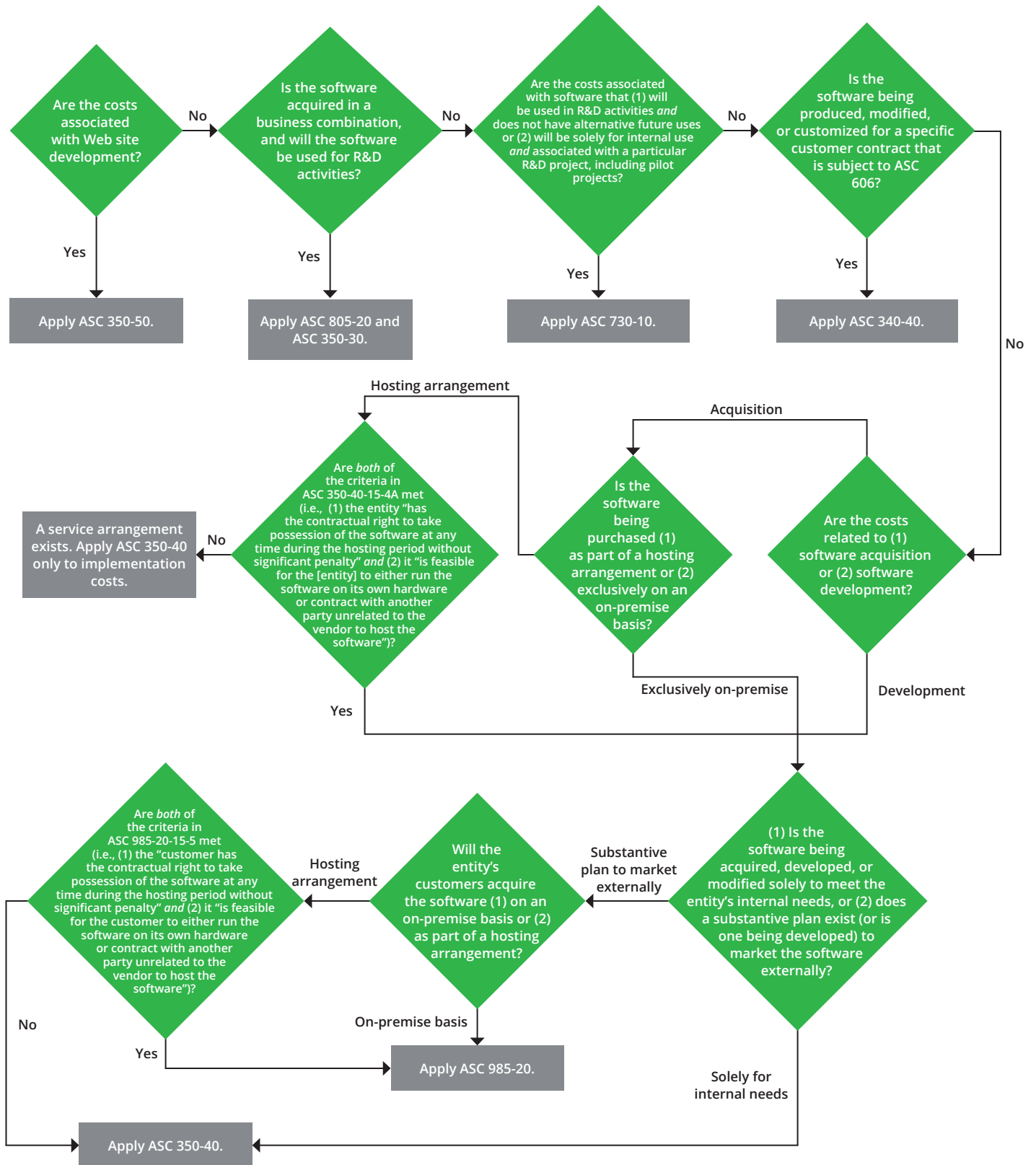
4.2.8 *Flowchart for Determining the Appropriate Guidance*

The flowchart below illustrates how an entity determines the appropriate guidance to apply to software-related costs.

¹⁰ See ASC 985-20-25-1.

¹¹ See ASC 350-40-15-7(b)(1) and ASC 730-10-55-1(h).

¹² See ASC 350-40-15-7(b)(2).



4.2.9 Importance of Ongoing Reassessment of Software Costs

As described above, there are various ways in which an entity's evolving business models may affect which guidance applies when accounting for costs to develop or acquire software. These include changes in the manner in which entities are (1) developing or acquiring software solutions from their vendors for internal use and (2) marketing and delivering software solutions to their customers. In the rapidly evolving technology ecosystem, it is important for an entity to have sufficient internal controls in place to periodically reassess and document how these changes in facts and circumstances may affect the guidance the entity should apply and the related accounting.

4.3 Accounting for the Development of Generative AI Software Products

4.3.1 Overview

More and more companies are considering purchasing or developing software that uses or leverages AI to enhance internal productivity or are incorporating generative AI into their revenue-generating products. Although developing or enhancing generative AI applications may involve more traditional software development costs (e.g., internal or external labor), some generative AI developments may be more advanced and may incur additional costs. For example, large amounts of data are typically necessary to train generative AI applications. Businesses that invest in generative AI will need to consider the accounting impacts of the software or software-related costs associated with generative AI.

The software costs incurred for generative AI could include fees paid to use third-party foundation models or large language models (LLMs) as well as fine-tuning and other training costs. Entities developing generative AI applications will need to consider whether the developed software will be used internally or whether it will be sold as a hosting arrangement, an on-premise software license, or a hybrid software offering. Depending on the nature of the generative AI software, the software costs related to the project may be subject to the guidance in ASC 985-20 or ASC 350-40 while other costs incurred to support the generative AI software may be subject to other U.S. GAAP.

The sections below discuss accounting considerations for entities developing generative AI technology (whether for their own use or external marketing). Specific topics addressed include (1) whether costs related to generative AI technology should be accounted for under ASC 350-30, ASC 350-40, ASC 985-20, or other U.S. GAAP and (2) how such costs should be evaluated under the applicable accounting literature.

4.3.2 Background

Generative AI is a subset of AI that focuses on the ability of machines to take in inputs (e.g., text, images) and create outputs in various formats (e.g., text, images, audio, code, voice, video). Generative AI applications serve as interfaces for end users. These applications are powered by significant infrastructure (e.g., cloud-based AI foundation models or LLMs) and generate content on the basis of how the underlying models were trained as well as the end user's inputs. Foundation models typically use neural networks to learn patterns from huge amounts of data and predict outcomes on the basis of historical data patterns. Like traditional AI, foundation models predict outputs by making inferences related to the inputs they receive. However, through fine-tuning, prompt engineering, and adversarial training (discussed further below), these models produce outputs on the basis of their understanding of human-generated inputs.

4.3.2.1 **Generative AI Foundation Model**

Generative AI applications are powered by foundation models, LLMs that use deep learning to process huge amounts of data. A foundation model can perform a wide range of tasks in natural language processing (NLP), a subfield of AI that enables computers to interpret input prompts and generate outputs such as text, translation, summarization, and answers to questions. With a foundation model, the software can predict outputs on the basis of statistical inferences it makes from the inputs received. The quality, accuracy, and relevancy of a generative AI application's outputs depend on the training the underlying foundation model receives.

The development of AI infrastructure foundation models is based on machine learning algorithms that consider linguistic rules and statistical models so that a computer can “understand” the natural language of a user's input prompt.

Entities that intend to develop a foundation model may incur the following costs:

- *Software developer costs* — Internal and external labor costs related to the development of the AI infrastructure code, time-intensive training, validation of outputs, and optimization activities.
- *Data acquisition costs* — Foundation model developers might incur costs to acquire the vast amount of data they need to develop foundation models that perform various NLP tasks and produce a wide range of responses.
- *Computation resources* — High-performance hardware must be used to train a foundation model. An entity could choose to purchase either specific hardware (e.g., computers with powerful CPUs, GPUs, or RAM) or platform or infrastructure services from third parties (i.e., cloud services) that offer the same computational scalability.
- *Storage costs* — Because of the amount of data needed to train and maintain a foundation model, entities may incur significant storage costs. An entity could choose to purchase or lease hardware for storage or to purchase cloud storage services from third parties.

Generative AI applications interact with and rely on a foundation model to generate outputs on the basis of user prompts. The dependency of AI applications on foundation models can be thought of as similar to the reliance of traditional software applications on operating systems. As a large-scale, pretrained language model, a foundation model serves as an engine that software developers train and calibrate for specific scenarios when creating their own generative AI applications. Currently, a limited number of companies have developed foundation models and some have provided access to this technology as open-source software or software sold as part of a hosting arrangement.

Because of the complexity of foundation models, the development of this technology is expected to be labor- and resource-intensive. We therefore expect most generative AI application developers to leverage existing foundation models in their applications rather than create their own. However, foundation models can vary in size and some entities might develop a private foundation model or LLM that is trained only on entity-specific data.

4.3.2.2 **Generative AI Applications**

Generative AI application developers are likely to incur additional training costs in refining foundation models to generate outputs tailored to their applications. Foundation models might be further trained through a combination of the following methods:

- *Fine-tuning* — Using specific data to train the foundation model to create outputs for a subset of prompts beyond the existing scenarios for which the model was trained. For example, a company that is creating a generative AI application to produce medical diagnoses for a user's symptoms may need to fine-tune the foundation model by acquiring information from medical encyclopedias, patient data, online databases of research articles, and scientific publications. An entity would expect to incur data acquisition and labor costs related to fine-tuning.
- *Prompt engineering* — Creating or adjusting the prompt to communicate with the foundation model to output an optimal answer. A company could incur specific internal or external software development costs in creating the prompt.
- *Adversarial training* — Two different deep-learning models can be pitted against each other to train both models. In this approach, one model, the generator, creates synthetic data samples while the other model, the discriminator, receives synthetic data samples and real data samples. The generator's objective is to produce samples that are indistinguishable from real data, while the discriminator's goal is to become better at distinguishing between real and generated data. An entity would expect to incur data acquisition, software development, and other labor costs related to adversarial training.

In addition to the data and training costs, an entity may also incur traditional software development costs (e.g., costs related to developing the software application user interface, infrastructure, graphics, and content) when creating a generative AI application. These costs may also be subject to capitalization or expense under ASC 985-20, ASC 350-40, or other U.S. GAAP.

4.3.3 **Accounting Considerations Related to Generative AI Development Costs**

Because generative AI is essentially a form of software, we believe that general software development accounting considerations apply to generative AI costs, including whether the related project will be used for internal purposes (including being sold as a service) or sold or marketed externally. Similar costs may be incurred, and similar considerations will be relevant, regardless of whether an entity is developing a foundation model (or LLM) or an application that leverages an existing model.

4.3.3.1 **Generative AI Software Used as Internal-Use Software or Software Marketed or Sold Only as a Hosting Arrangement**

Although the development of foundation models marked a crucial milestone in NLP and AI research, there was initially significant uncertainty about whether this technology would meet its specified performance requirements. Development risks affect whether costs incurred to develop software can be capitalized. Different guidance applies depending on whether the software (e.g., a foundation model or application) is being developed for internal use or for external sale or marketing. Entities developing AI software for internal use or to be sold or marketed as a hosting arrangement would consider applying the capitalization guidance in ASC 350-40 on internal-use software. Alternatively, ASC 985-20

would apply if an entity intends to sell or market its AI software as software licenses. With respect to internal-use software, ASC 350-40-15-2A states the following:

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.

When AI software is developed for internal use, a significant portion of the software development costs may be incurred during the preliminary project stage. Unlike ASC 985-20 (discussed further below), ASC 350-40 does not require the establishment of technological feasibility¹³ for capitalization but does have other requirements for capitalization depending on the stage of development. Generally, development costs incurred during the application development stage are capitalized, while costs incurred during the preliminary project stage and postimplementation-operation stage are expensed as incurred.

The following are some indicators of when an entity is in the preliminary project stage and when costs should therefore be expensed as incurred in accordance with ASC 350-40:

- The entity is considering allocating resources (i.e., developers, financial budget) between different projects (e.g., different AI applications or other software projects).
- The entity is still determining the performance requirements for the AI application or the infrastructure requirements necessary for the application to operate.
- The entity is holding ongoing conversations with vendors (e.g., foundation model vendors, hardware vendors, cloud computing vendors) to determine which products are best aligned with the entity's software performance requirements.
- The entity is still exploring alternatives related to achieving the performance requirements identified (i.e., using internal software developers to train foundation models versus hiring third-party consultants).
- The entity is determining whether there is an existing technology for developing a specific generative AI application to meet the identified performance requirements.

The preliminary project phases for many AI applications and software projects may be longer than those for other software development projects given the use of new and advanced technologies as well as the emergence of high-risk development issues that could affect the successful completion of the project.

Once the preliminary project stage is complete and the application development phase commences, entities developing AI applications will need to identify and capitalize the direct internal and external costs incurred to develop the AI application. When both foundation models and AI applications are being developed or implemented at the same time, entities will need to carefully track internal and external costs to ensure they are appropriately deferred or capitalized in accordance with the guidance in ASC 350-40. In addition, the development and implementation of foundation models and AI applications may involve new activities (e.g., training) performed by employees who have not historically tracked time spent on developing software. In these circumstances, entities may need to create new processes and controls to track these costs accurately.

¹³ ASC 350-40 stipulates that to proceed from the preliminary project stage to the application development stage, a company would have to determine that the technology it needs to meet the performance requirements exists.

4.3.3.2 *Generative AI Software That Will Be Sold or Marketed Externally*

If an entity plans to license its generative AI software externally, the software would be within the scope of ASC 985-20. The costs of developing software within the scope of ASC 985-20 cannot be capitalized until technological feasibility is established, which typically occurs toward the end of the development period when all high-risk development issues have been resolved through coding or testing. ASC 985-20-25-1 states the following:

ASC 985-20

25-1 All costs incurred to establish the technological feasibility of a computer software product to be sold, leased, or otherwise marketed are research and development costs. Those costs shall be charged to expense when incurred as required by Subtopic 730-10.

As a result, minimal costs tend to be capitalized when software is developed to be marketed or sold externally unless the costs incurred are subject to other GAAP.

4.3.3.3 *Generative AI Software Acquired as a Cloud Computing Arrangement*

Rather than develop AI software for internal use, an entity may engage with a third party to develop an AI solution that will be accessed as part of a cloud computing arrangement. In such circumstances, the AI software will only be accessed as part of a hosting arrangement. The ASC master glossary defines a hosting arrangement as follows:

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

Under ASC 350-40, costs incurred to implement a hosting arrangement that is a service contract would be subject to the same recognition and measurement guidance as costs incurred to develop or acquire internal-use software. However, any costs deferred in accordance with this guidance would be presented in the same manner as any prepayments made for the underlying service.

4.3.3.4 *Data Acquisition Costs*

As noted above, entities developing foundation models and AI applications may need significant amounts of data to train the models. Entities will need to consider whether the costs of acquiring the data should be (1) expensed as incurred, (2) recognized as a separate intangible asset, or (3) considered for capitalization as part of the AI application or foundation model.

Costs incurred to acquire data from a third party should be evaluated to determine whether it is appropriate to capitalize the costs as a separate intangible asset. The guidance in ASC 350-30 would apply to data that are acquired individually or as a group of other assets (that do not constitute a business) and that have an alternative use (i.e., in more than one software project). ASC 350-30-25-1 states that “[a]n intangible asset that is acquired either individually or with a group of other assets shall be recognized.” Further, ASC 350-30-25-4¹⁴ states the following regarding the acquisition of intangible assets:

ASC 350-30

25-4 Intangible assets that are acquired individually or with a group of assets in a transaction other than a business combination or an acquisition by a not-for-profit entity may meet asset recognition criteria in FASB Concepts Statement No. 5, *Recognition and Measurement in Financial Statements of Business Enterprises*, even though they do not meet either the contractual-legal criterion or the separability criterion (for example, specially-trained employees or a unique manufacturing process related to an acquired manufacturing plant). Such transactions commonly are bargained exchange transactions that are conducted at arm’s length, which provides reliable evidence about the existence and fair value of those assets. Thus, those assets shall be recognized as intangible assets.

Pending Content (Transition Guidance: ASC 805-60-65-1)

25-4 Intangible assets that are acquired individually or with a group of assets in a transaction other than a business combination, an acquisition by a not-for-profit entity, or a joint venture upon formation may meet asset recognition criteria in FASB Concepts Statement No. 5, *Recognition and Measurement in Financial Statements of Business Enterprises*, even though they do not meet either the contractual-legal criterion or the separability criterion (for example, specially-trained employees or a unique manufacturing process related to an acquired manufacturing plant). Such transactions commonly are bargained exchange transactions that are conducted at arm’s length, which provides reliable evidence about the existence and fair value of those assets. Thus, those assets shall be recognized as intangible assets.

Pending Content (Transition Guidance: ASC 105-10-65-9)

25-4 Intangible assets that are acquired individually or with a group of assets in a transaction other than a business combination, an acquisition by a not-for-profit entity, or a joint venture upon formation may qualify for recognition even though they do not meet either the contractual-legal criterion or the separability criterion for being an identifiable asset (for example, specially-trained employees or a unique manufacturing process related to an acquired manufacturing plant). Such transactions commonly are bargained exchange transactions that are conducted at arm’s length, which provides reliable evidence about the existence and fair value of those assets. Thus, those assets shall be recognized as intangible assets.

Acquired data will lack physical substance and will most likely be acquired as part of a contract that defines the rights controlled by the entity. In these cases, the acquired data are likely to meet the definition of an asset (because the data are separately identifiable and provide an entity with a present right to future economic benefits) and could be recognized separately as an intangible asset. Entities would need to determine the useful life of the acquired data and perform an impairment assessment in accordance with ASC 350.

¹⁴ The amendments in [ASU 2023-05](#) — which are effective prospectively for all joint venture formations with a formation date on or after January 1, 2025 — add to the exclusions in ASC 350-35-25-4 intangible assets acquired individually or with a group of assets in a joint venture upon formation. In addition, the amendments in [ASU 2024-02](#) — which are effective for fiscal years beginning after December 15, 2024, for public business entities and fiscal years beginning after December 15, 2025, for all other entities — remove all Concepts Statement references from the Codification. However, we do not believe that the removal of the reference to Concepts Statement 5 from ASC 350-30-25-4 will affect the application of the guidance in this paragraph to data acquisition costs.



Connecting the Dots

The existing guidance in ASC 350-30-25-4 refers to the asset recognition criteria in [FASB Concepts Statement 5](#). However, as noted in footnote 14, ASU 2024-02 removed the references to the Concepts Statements throughout the Codification. Further, the definition of an asset in FASB Concepts Statement 5 was amended by FASB Concepts Statement 8. The definition of an asset in paragraph E17 of [FASB Concepts Statement 8, Chapter 4](#), is as follows:

An asset has the following two essential characteristics:

- a. It is a present right.
- b. The right is to an economic benefit.

A present right of an entity to an economic benefit entitles the entity to obtain this benefit from the right and to restrict others' access to it. We believe that rights to data acquired from a third party would generally meet the definition of an asset. Further, while there are differences in the definition of an asset under the two Concepts Statements, the differences are not expected to significantly change what does and what does not represent an asset. Accordingly, we believe that rights to data acquired from a third party would generally meet either definition of an asset.

Although costs incurred to acquire data from a third party would generally be capitalizable as an intangible asset, data acquisition costs would be expensed as incurred under ASC 730-10 if the data will be used in R&D activities *and* do not have alternative future uses.¹⁵ Such data costs would include those incurred for a specific software development project that is within the scope of ASC 985-20 for which technological feasibility has not been established. This is because, as noted above, costs incurred to develop technological feasibility are considered R&D activities within the scope of ASC 730-10. ASC 730-10-25-2 states, in part:

Elements of costs shall be identified with research and development activities as follows (see Subtopic 350-50 for guidance related to website development): . . .

- c. Intangible assets purchased from others. The costs of intangible assets that are purchased from others for use in research and development activities and that have alternative future uses (in research and development projects or otherwise) shall be accounted for in accordance with Topic 350. The amortization of those intangible assets used in research and development activities is a research and development cost. **However, the costs of intangibles that are purchased from others for a particular research and development project and that have no alternative future uses (in other research and development projects or otherwise) and therefore no separate economic values are research and development costs at the time the costs are incurred.** [Emphasis added]

Data may also be acquired for a specific software project that is being developed for internal use and does not have an alternative future use (e.g., other software projects). In this case, rather than being a separate intangible asset, the data costs may be direct external costs incurred to develop internal-use software within the scope of ASC 350-40. Specifically, an entity could purchase data to train generative AI applications, resulting in the creation of new functionalities. If the AI software project is in the application development stage, it may be appropriate to capitalize the data acquisition costs as direct costs incurred during that phase. Alternatively, as discussed further below, if the data and resulting training were only necessary to maintain the existing features or functionality of the generative AI application, capitalization would not be appropriate because the costs would be akin to maintenance costs. Further, any costs incurred in the preliminary project phase of development should be expensed as incurred.

¹⁵ The term "alternative future use" is not defined in U.S. GAAP. However, Section 3.14 of the AICPA Accounting and Valuation Guide [Assets Acquired to Be Used in Research and Development Activities](#) states, "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)" (footnote omitted).

If acquired data have an alternative future use (as discussed above) and are separately recorded as an intangible asset in accordance with ASC 350-30, we do not believe that the subsequent amortization of the intangible asset would be included as a cost eligible for capitalization under the internal-use software guidance. In such circumstances, the subsequent amortization would not be considered a direct cost incurred during the application development stage and would therefore not be within the scope of ASC 350-40.¹⁶

4.3.3.5 Upgrades and Enhancements

After the initial release of their generative AI software, entities will most likely improve the functionality of their application through additional software development and fine-tuning. An entity that develops AI software for internal use should consider whether incurring these costs is associated with an upgrade or enhancement to internal-use software as described in ASC 350-40-25-7 through 25-9:

ASC 350-40

25-7 Upgrades and enhancements are defined as modifications to existing internal-use software that result in additional functionality — that is, modifications to enable the software to perform tasks that it was previously incapable of performing. Upgrades and enhancements normally require new software specifications and may also require a change to all or part of the existing software specifications. In order for costs of specified upgrades and enhancements to internal-use computer software to be capitalized in accordance with paragraphs 350-40-25-8 through 25-10, it must be probable that those expenditures will result in additional functionality.

25-8 Internal costs incurred for upgrades and enhancements shall be expensed or capitalized in accordance with paragraphs 350-40-25-1 through 25-6.

25-9 Internal costs incurred for maintenance shall be expensed as incurred.

Upgrades and enhancements to generative AI applications that are sold or marketed externally, and that are within the scope of ASC 985-20, would be subject to the same capitalization threshold as the initial product development (i.e., an entity is required to establish technological feasibility of the upgrade or enhancement to capitalize associated costs).

Maintenance activities would be expensed as incurred for all software. ASC 350-40 does not define the term “maintenance,” but ASC 985-20-20 defines it as follows:

Activities undertaken after the product is available for general release to customers to correct errors or keep the product updated with current information. Those activities include routine changes and additions.

A key consideration related to incurring data costs to train the AI software after initial deployment is whether additional training results in the creation of new functionality (e.g., whether the AI application can perform a different task) or whether ongoing training is necessary to retain the relevance of the AI application (e.g., maintain its intended functionality). Data and associated training that are intended to keep an AI application current or relevant would most likely be considered maintenance. Unless the costs are separately capitalizable as an intangible asset, such costs would be expensed as incurred.

¹⁶ ASC 350-40-30-1 states that the only internal-use software costs that would be capitalized include (1) “[e]xternal direct costs of materials and services consumed in developing or obtaining internal-use computer software”; (2) “[p]ayroll and payroll-related costs . . . 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”; and (3) “[i]nterest costs incurred while developing internal-use computer software.”

In contrast, training that creates new functionality might be considered an upgrade or enhancement. Therefore, entities will need to determine whether the additional fine-tuning they are performing maintains the current software features of their generative AI application or whether the fine-tuning introduces additional software features that did not previously exist. This would dictate whether the data costs incurred to perform the fine-tuning should be capitalized as costs incurred to develop a software upgrade, expensed as software maintenance, or evaluated for capitalization separately as an intangible asset.

4.3.3.6 Computation Resources and Storage Costs

In supporting generative AI applications and the underlying foundation or LLM model, an entity may incur significant costs related to (1) hardware for computation resources and (2) storage costs. Generally, such costs will be accounted for under U.S. GAAP other than ASC 985-20 and ASC 350-40. Servers, computers, GPUs, and CPUs purchased to increase an entity's computational power and build out its storage infrastructure would be accounted for as long-lived assets under ASC 360.

If an entity enters into a hosting arrangement with a vendor to leverage the vendor's computation or storage capabilities, it is likely that the arrangement will be accounted for as a service arrangement. Typically, in such circumstances, (1) the entity does not have "the contractual right to take possession of the software at any time during the hosting period without significant penalty"¹⁷ or (2) it is not "feasible for the [entity] to either run the software on its own hardware or contract with another party unrelated to the vendor to host the software."¹⁸ Accordingly, the costs incurred to implement third-party infrastructure or storage services would be evaluated for capitalization in accordance with ASC 350-40 and, if capitalized, would be deferred as a prepaid asset and recognized over the contract period (as well as over periods for which contractual renewals are reasonably certain to be exercised). Note that ongoing costs to use or maintain the third-party infrastructure or storage services would not meet the deferral criteria.

In addition, as noted above, AI applications typically need to be developed to work with one or more AI foundation or LLM models. An entity that enters into a hosting arrangement with a vendor to purchase a foundation model will need to determine whether it has (1) purchased or licensed software or (2) purchased a service arrangement. The entity must perform this assessment regardless of whether the foundation model will be used to create a generative AI application for internal use or whether it will be sold as a hosting arrangement or an on-premise license. We expect that most entities will determine that the foundation models or LLMs they acquire will be through a service contract, which could be accounted for as a prepaid asset if an up-front payment is made for the future use of the functionality.

4.4 On the Horizon — Proposed ASU on the Accounting for and Disclosure of Software Costs

In October 2024, the FASB issued a [proposed ASU](#) that would amend certain aspects of the accounting for and disclosure of software costs under ASC 350-40. Rather than revising the guidance on this topic in its entirety, the Board is proposing targeted improvements to address specific issues raised by stakeholders. In addition, the proposed ASU would not amend the cost guidance for software licenses that are within the scope of ASC 985-20.

¹⁷ ASC 985-20-15-5(a).

¹⁸ ASC 985-20-15-5(b).

To clarify that the guidance applies to both linear and nonlinear software development, the proposed ASU would remove all references to “development stages” from ASC 350-40 as well as amend the threshold for the capitalization of costs. Under current GAAP, capitalization of software development costs for internal-use software is required once the preliminary project stage is complete. The proposed ASU instead provides the following two criteria in ASC 350-40-25-12 that must be met for entities to begin capitalizing software costs:

- “Management, with the relevant authority, implicitly or explicitly authorizes and commits to funding a computer software project.”
- “It is probable that the project will be completed and the software will be used to perform the function intended (referred to as the probable-to-complete recognition threshold).”

Under the proposed ASU, the probable-to-complete threshold would not be met when there is “significant uncertainty associated with the development activities of the software (referred to as significant development uncertainty).” The proposed guidance provides the following factors that may be indicative of significant development uncertainty:

- “The software being developed has novel, unique, unproven functions and features or technological innovations.”
- “The significant performance requirements of the computer software have not been identified or the significant performance requirements continue to be substantially revised.”

For more information about the proposed ASU, see Deloitte’s November 5, 2024, [Heads Up](#).

4.5 SEC Comment Letter Trends

The SEC staff will occasionally issue comments related to software development costs to registrants in the technology industry. For example, if a registrant only sells software as SaaS but has not capitalized any software costs, the SEC staff may ask the registrant to explain why no development costs were capitalized. The staff may also ask for more details about the development process for internal-use software, recently developed offerings, added functionality, and what consideration was given to disclosing policies related to such software. In addition, the staff may ask questions regarding the income statement classification and the description of any expenses related to software development (e.g., amortization expense or expensed development costs).

For more information, see [Section 6.5.1.1](#) of Deloitte’s Roadmap [SEC Comment Letter Considerations, Including Industry Insights](#).