Internal use software is an item that is often overlooked when developing asset capitalization policies. Further, this is an area where those with less accounting experience may get tripped up in the nuances found within the accounting codification. Not properly following accounting guidance during periods where there is no independent third party review or audit can lead to the need for large restatements of prior reporting periods as the development of software may occur over several reporting periods. This article will present a clear methodology to follow when expensing versus capitalizing expenditures for internal use software, by applying the principles established in the Accounting Standards Codification 350-40, Internal Use Software.

The first step is to determine whether or not you actually have internal use software. In order to qualify for this category, the software must meet both of the following criteria:

1. The software must be acquired, internally developed, or modified solely for the purpose of meeting internal needs.
2. During the development or modification period, there cannot be a substantive plan currently in existence or being developed to market the software externally.

We come across our first nuances within these two criteria. What is defined as “meeting internal needs” and what exactly is a “substantive plan”? While “meeting internal needs” appears straightforward, and likely is in most circumstances, situations do appear that can make this seemingly straightforward term appear vague. Consider the following facts: your software is used in the production of your product or while providing a service, but your customer does not acquire the software or the future right to use it. Is the software still considered to be used for meeting internal needs? The answer is “yes” and hinges on what product or service is actually being consumed by the customer. Consider the following example for clarity; a communications company selling telephone services has software included in the telephone switch that is part of the internal equipment used to deliver the service, but is not part of the product or service actually being acquired or received by the customer and is therefore considered internal use software. This step is important because, if the software fails to be used for internal needs, then it is likely that the guidance of ACS 985-20, Costs of Software to be Sold, Leased, or Marketed, may apply instead. The second nuance of the stated criteria is determining what qualifies as a “substantive plan.” A substantive plan would include the selection of a marketing channel with identified promotional, delivery, billing, and support activities. Also, it must be reasonably possible that the plan could be implemented.

Once you have determined that you have internal use software, you must determine which stage of development you are in. The stage of development will determine whether or not costs should be expensed or capitalized.

The initial stage is considered to be the “preliminary project stage.” All internal and external costs during this period should be expensed as incurred. During this stage companies are in the process of determining whether or not to pursue the development of software, but no actual activities of designing the software have actually begun.

When a software project is in the preliminary project stage, entities will likely experience the following:

- Strategic decisions will be made to allocate resources between alternative projects at a given point in time.
- Performance and system requirements will be determined for the proposed project (e.g., determining what it is that the software is needed to do and what problem it is to solve.)
- Receive demos from vendors and explore alternative methods to solving the stated problem.
Determine if such a technology exists.

Select a vendor if externally produced or seek a consultant to assist in the development or installation of the software.

The next and most substantial stage is called the “application development stage.” At this point, management has implicitly or explicitly authorized and committed to funding a computer software project, it is probable that the project will be completed and the software will be used to perform its intended function. Another small nuance appears: what is considered to be authorization? Authorization can be in the form of an execution of a contract with a third party to agree to develop the software, approval of the expenditures related to internal development, or commitment to purchase the software from a third party. During this stage expenditures can be both capitalized and expensed, depending on the nature of the transactions. Development costs are capitalized along with any costs to develop or obtain software that permits access to or conversion of previous system data by new systems. All other data conversion costs are expensed as incurred along with training costs during this stage of development. Costs that can be capitalized include external direct costs of materials and services consumed during development, payroll and related costs (e.g., 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 directly spent on the project (i.e., design, coding, installation to hardware, testing), and interest costs incurred while developing internal use software (in accordance with the provisions of the ASC 835-20). Toward the end of the application development stage, there comes a time when the software is considered to be substantially complete and ready for its intended use. In other words, all substantial testing is complete. When this occurs, no further costs should be capitalized and you have now successfully navigated your way through the preliminary and the application development stage and are ready to move to the final stage. If, however, during this stage it is determined that a project is no longer probable to be completed and placed into service, then no further costs are to be capitalized and impairment testing will occur.

Software that has been successfully developed as a solution to a stated problem and is now up and running brings us into the “post implementation-operation stage.” Internal and external training costs, along with maintenance are still going to be expensed as incurred; however, now there is the nuance of the treatment for upgrades and enhancements. If it is probable that the upgrades and enhancements will result in additional functionality, then those costs are capitalized. If maintenance and relatively minor upgrades and enhancements cannot be separated on a reasonably cost-effective basis, then all of the expenditures should be expensed as incurred. Upgrades and enhancements should only be capitalized if the costs can be segregated. What about when existing software is replaced with new software? When this occurs, the unamortized costs of the old software should be expensed at the time that the new software is ready for its intended use.

Expenditures were either expensed or capitalized during the three stages of development and amortization of those expenditures must be considered. First, a useful life must be determined. Issues like obsolescence, technology, competition, and other economic factors must be considered when determining the useful life of the internal use software. Technology has a history of rapidly changing and, as such, software often has a relatively short useful life. Once the useful life is determined, the amortization methodology must be determined. Straight-line amortization is the most commonly-used technique, however; alternative methods are considered acceptable if rationale for the basis better represents the use of the software. Again, another nuance appears: when should the capitalized expenditures begin to be amortized? Should this occur after capitalization, when the software is put into service, or at some point during development? The answer is that amortization should begin when the software is ready for its intended use, which is when all substantial testing is complete. This does not change even if the software will be placed into service in planned stages.
The last issue that companies with internal use software should consider is impairment. As aforementioned, this should be evaluated when it is determined that the project is not expected to provide substantive service potential. Other situations may also arise that will require management to evaluate if impairment has occurred, such as the following:

› There is a change in the manner of expected use.
› There is a significant change made or to be made to the software.
› Costs of the project begin to significantly exceed the original budgeted amount.

When it is no longer probable that the project will be completed and placed in service, or when the software is no longer effectively serving its stated purpose, then the asset will be reported at the lower of the carrying amount or fair value, if any, less costs to sell. Usually, the software at this moment is presumed to have zero fair value and any unamortized value is then expended.

The process of capitalizing versus expensing costs related to internal use software appears straightforward on the surface, but many nuances may arise. However, understanding which stage of development the software is in and using this article as a guide will result in proper compliance with the accounting codification and mitigate against the risk of potentially having material misstatements that could result in prior period adjustments.

Please contact Keiter with any questions regarding the asset capitalization of internal use software at 804.747.0000 | information@keitercpa.com.