



Software capitalization:

Accounting for software development cost in the age of cloud and agile

IT innovation is showing its mettle. For most industries, Deloitte anticipates investments in innovation and business growth to return to prepandemic levels by 2022.¹ As though to underscore the sentiment, Gartner recently projected an uptick of 9% in global IT spending for 2021.²

Along the way, many technology companies are transitioning to agile development and cloud delivery models. Both offer advantages that enable organizations to operate more efficiently. At the same time, however, accounting for software development cost can be trickier in an agile or cloud environment, raising the risk of a financial statement error or a failure to capitalize sufficient costs (which has

an immediate effect on reported net income). The nuances of the challenge are different between cloud and agile. The following are some key considerations:

Agile development techniques

To understand the issues associated with agile development, consider that the accounting standard for software capitalization was written for the way organizations historically developed software. Older, so-called waterfall development methods were linear. They had clear delineations between phases, from up-front planning to release and ongoing maintenance, which internal accounting policies and systems could readily reflect.

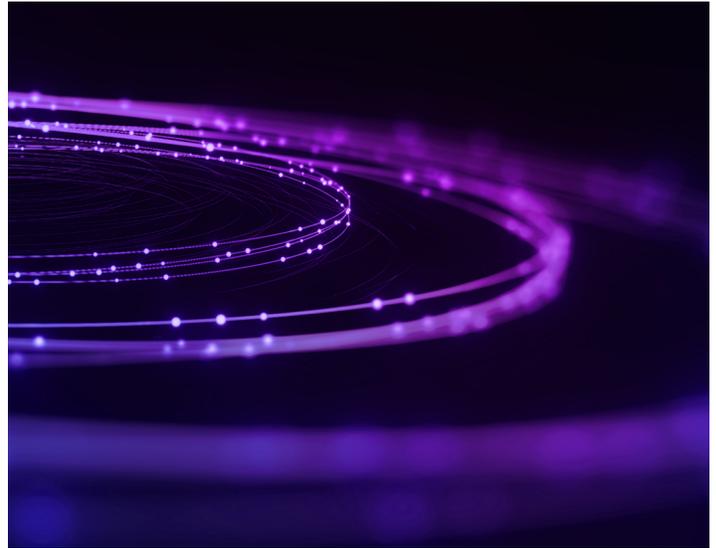
Then companies shifted to more agile methods of software development. Agile development teams are organized by product and task rather than by overall project. In addition, the teams themselves can be highly fluid, with people often staying only as long as their skills are required. These and other process innovations help software developers produce functionality in a faster, nimbler way.

But the same processes that make agile development teams more efficient can also make it harder to identify costs for capitalization. More specifically, because agile efforts are fluid, it can be difficult to differentiate development from planning and maintenance, since agile teams often go through all three phases very quickly during a sprint. Further, it can be difficult to identify sprints that include efforts that qualify for capitalization.

It helps if an entire product team is dedicated to application development. However, if the product team or scrum is multidisciplinary—not unusual in the world of agile—the accounting team will have to learn what individual team members customarily do, then allocate the costs accordingly (say, by capitalizing a percentage of the team's cost during a specific time frame).

And then there's story abandonment. One of the criteria to capitalize costs is that management has approved the project and the funds have been committed to complete development. But if the organization has a history of abandoning sprints or stories before releasing them into production, it calls into question whether those costs are really for development and therefore eligible to be capitalized at all.

Organizations can also hit a snag when determining costs related to identified enhancements. When an agile team develops functionality into production, they may or may not have created a new asset that the organization should start amortizing. It depends on whether the team intends to build on their work in a subsequent sprint. And if the new asset really is complete, that means other costs may have to be abandoned or impaired, and it's not always clear which costs get what kind of treatment. All of this can make it hard to properly identify and capitalize costs in an agile environment.



Cloud delivery models

The shift to cloud delivery models means companies are developing software to provide a service versus software to be marketed or sold as a product (like a traditional software license sold as an on-premises solution).

The distinction is important because software capitalization requirements are different between the two. For software that the organization aims to sell or market, most (if not all) of the development cost is expensed as incurred. For software that the organization will deliver as a service, on the other hand, much of the development cost will likely have to be capitalized. Although these guidelines seem straightforward enough, the timing of a shift from on-premises to cloud delivery may not always be clear.

The accounting gets especially complicated when the organization delivers software through a hybrid of cloud and on-premises infrastructure. This is because an entity will need to identify development activities that are dedicated to the cloud environment, which may need to be accounted for differently from the activities that enhance the on-premises technology. On top of that, the accounting team may lack visibility into a few vital pieces of information needed to properly identify and account for development costs. One of those pieces is when functionality is no longer available for on-premises deployment (which could be the case when new functionality is only available in the cloud). Another is which planning, application development, and maintenance costs are unique to the cloud functionality and which costs relate to on-premises functionality.

A common language between finance and IT

The takeaway is that cloud delivery models and agile development techniques each have unique accounting considerations and impacts. Together, they can significantly increase process complexity for the organizations that adopt them.

So where can finance, accounting, and IT leaders go from here? Start by taking the following steps:



Understand what information the development team produces, and to what level of detail.



Compare the development team's information with the information necessary to appropriately account for the organization's technology spend.



Collaborate with developers to identify starting and stopping points for stories, epics, or other units of account consistent with their development discipline (see sidebar).



Close the information gap in a way that's meaningful to both development and accounting (such as tagging activities and creating a process to extract tags for determining costs eligible for capitalization).



Evaluate the utility of information in making decisions about technology spend, and work with stakeholders to close any gaps there.

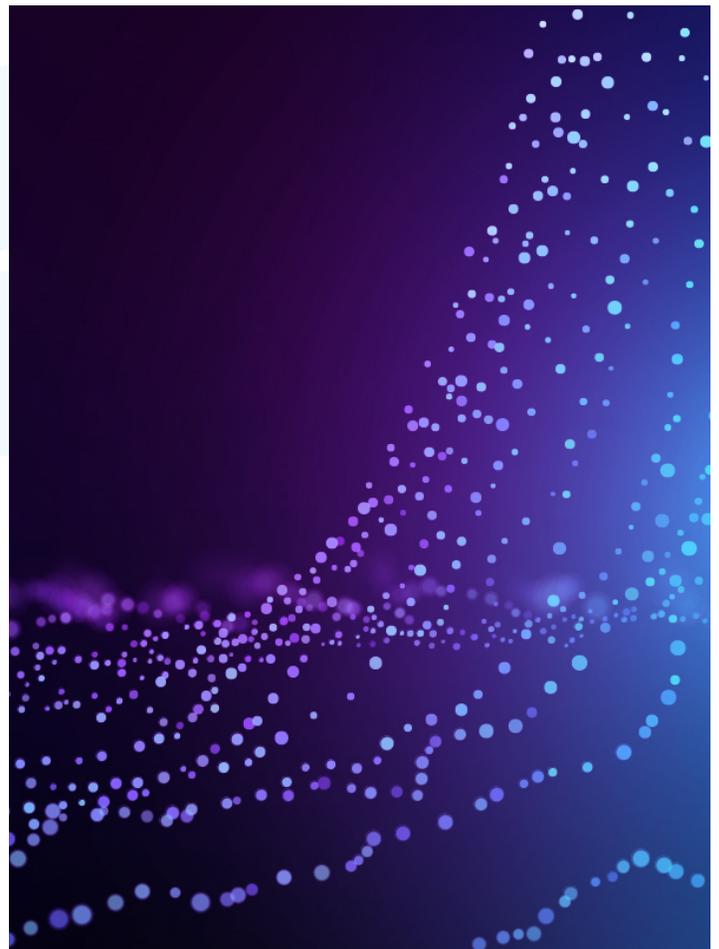
Cloud computing and agile development lie at the heart of many digital transformation efforts, offering organizations speed and flexibility alongside the potential for continuous innovation. The flip side is that they can also make accounting for software development costs significantly more complex. By working together, IT and finance can craft a common language that brings insight and control to the application development phase of a product development life cycle.

This article touches on the broader challenges of capitalizing software in an agile or cloud environment. However, every organization has its own complexities to consider. With help from Deloitte, you can create an approach that's effective and practical for both accounting and development. Please contact the authors listed on the next page or your local Deloitte office representative if you'd like to discuss further.

What is agile software development?

Agile is a methodology for putting usable new software in people's hands as quickly as possible. The basic approach is to break projects up into small, time-limited chunks called sprints. Each sprint addresses at least one user requirement, or story, quickly resulting in a working prototype. (An epic is a series of user stories.)

Users get frequent demos of the software as it is being developed, with a chance to offer their feedback to the development teams. The teams then generate a new version that incorporates this user feedback. This process is repeated until the functionality is right.



Software capitalization: Accounting for software development cost in the age of cloud and agile

Contact us



Chris Chiriatti

Managing Director, Audit & Assurance
Deloitte & Touche LLP
cchiriatti@deloitte.com
+1 203 761 3039



Sean Torr

Managing Director, Risk & Financial Advisory
Deloitte & Touche LLP
storr@deloitte.com
+1 615 259 1888

Endnotes

1. Khalid Kark, Tim Smith, and Jagjeet Gil, "[Maximizing the impact of technology investments in the new normal](#)," *Deloitte Insights*, February 3, 2021.
2. Gartner, "[Gartner forecasts worldwide IT spending to grow 9% in 2021](#)," July 14, 2021.



<http://www.deloitte.com/us/accounting-advisory-transformation>

The services described herein are illustrative in nature and are intended to demonstrate our experience and capabilities in these areas; however, due to independence restrictions that may apply to audit clients (including affiliates) of Deloitte & Touche LLP, we may be unable to provide certain services based on individual facts and circumstances.

This article contains general information only and Deloitte is not, by means of this article, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This article is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional adviser. Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

As used in this document, "Deloitte" means Deloitte & Touche LLP, which provides audit, assurance, and risk and financial advisory services, which provides advisory services. These entities are separate subsidiaries of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting.

Copyright © 2021 Deloitte Development LLC. All rights reserved.