**ACCOUNTING FOR AGILE SOFTWARE DEVELOPMENT**

by Terry Quan and Pat Reed

June 3, 2013

Over the past two decades, internally developed software has become an increasingly critical part of operations for most organizations, essential for many core functions: financial reporting, supply chain management, customer engagement, etc. Since the *Manifesto for Agile Software Development* in 2001, software development methodology has steadily changed from a phased, waterfall model to the agile model. In the agile model, teams work in iterations of planning, designing, building and testing as requirements often change even when a project’s business goals remain the same. This method is in contrast to the linear, phased approach of the waterfall method. Because of this difference, iterative versus phased, it is worth clarifying the accounting of agile software development expenditures and particularly when to capitalize them.

The current accounting standard on internally developed software is FASB's Accounting Standards Codification (ASC) 350-40 *Internal Use Software,* which codifies AICPA's Statement of Position (SOP) 98-1 *Accounting for the Costs of Computer Software Developed or Obtained for Internal Use*. ASC 350-40 and SOP 98-1 fit well with the waterfall approach to software development, which was the common methodology at the time SOP 98-1 was written. In the waterfall method of software development, tasks are readily organized into the three discrete, linear phases of SOP 98-1: (1) analysis of all customer requirements, assessments and planning, (2) development, installation and testing, and (3) post-implementation and operation. In phases (1) and (3), costs are expensed. In Phase (2), overhead costs are expensed, and development costs are capitalized.

Agile software development, the method most popularly used today, is more compatible with dynamic business environments because it reduces the cost of change and rapidly delivers the highest-valued features to the customer through collaboration, rapid iterations, and re-prioritization. The agile method readily adapts to emerging business priorities. However, accounting for software development expenditures is more challenging with this method. Accountants may be unclear when to capitalize agile-developed software projects. Consequently, conservative accountants may expense all project labor costs related to agile projects or require development teams to keep detailed daily time sheets.

The former practice biases against the use of agile, and the latter is inefficient and impractical. The increasing use of agile software development necessitates clear guidance for the capitalization of expenditures. This is important for appropriate capital allocation and consistency, both within the firm and across firms. Agile Accounting was initiated to support the community of agile-method developers and understand the treatment of expenditures related to agile projects under Generally Accepted Accounting Principles in the United States.

**The Increasing Importance of Internal Software**

Software for internal use has become a significant asset for many companies. FASB's ASC 350-40 codifies SOP 98-1 *Accounting for the Costs of Computer Software Developed or Obtained for Internal Use,* which was written in 1998 when internally developed software assets were less significant for most non-technology companies. Today, even companies that are not in the business of selling software have significant software assets, as indicated by the footnotes in the Property, Plant and Equipment (PP&E) section of their recent 10-K filings with the Securities and Exchange Commission (SEC). One example, Gap Inc.’s 2012 SEC 10-K shows gross in-house software assets of over a billion dollars. While companies generally do not provide details about accumulated depreciation of PP&E, it is clear that the amount of capitalized software is significant and necessitates consistent capitalization of software development expenditures. Typically, capitalized software fully depreciates after 3 years.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Software Assets as Reported in SEC 10-Ks of Sample Firms (2012)** | | | | |
| **Company** | **Fiscal Year End** | **PP&E**  **Net** | **Software Assets**  **Gross** | **PP&E**  **Gross** |
| Gap Inc. | 02/02/13 | $2,619.00 | $1,078.00 | $7,910.00 |
| Levi Strauss | 11/25/12 | $458.81 | $285.96 | $1,241.57 |
| Nordstrom | 02/02/13 | $2,579.00 | $518.00 | $6,643.00 |
| Ralph Lauren | 03/30/13 | $932.20 | $252.10 | $2,253.50 |
| Williams Sonoma | 02/13/13 | $812.04 | $366.51 | $2,094.94 |
| Source: Securities and Exchange Commission 10-Ks. All amounts in millions. | | | | |

**Accounting for Agile Project Labor Costs**

To understand the capitalization of agile development expenditures, we should return to the definition of an “asset” and “capitalization,” and as well recall the spirit of ASC 350-40 and SOP 98-1. The three stages of an IT project outlined in SOP 98-1 can be applied to agile as well as waterfall-developed software projects. An asset is defined as a potential future economic benefit that the firm controls based on past transactions. To capitalize is to record an expenditure as an asset rather than to treat it as an expense of the current period. The following describes the three different stages for an agile-developed project:

1. The Preliminary Project Stage includes conceptual formulation, evaluation and final selection of alternatives. Generally, once technical feasibility and the business case analysis are completed, the project is ready for consideration for funding approval by a company representative with budget authority. After approval, the project team participates in a kick-off, or project inception, to deepen their understanding of WHAT the customer needs and to ensure that they will deliver an economic benefit to the organization with a high-level release plan. All costs are expensed at this stage.
2. In the Application Development Stage, the design of the chosen path, including software configurations and interfaces, marks the capitalization of project labor costs. The team now moves from WHAT to HOW. All work associated with designing, developing, coding, testing and installing of hardware are valuable features for the customer that can be capitalized. The customer may even be an internal department which will indirectly benefit the company’s net cash flows in the future. One exception at this stage is administrative overhead, training and manual data conversion which should be expensed.
3. The Post-Implementation / Operation Stage begins after the code is in production and after final customer acceptance, testing and stabilization are completed. Typical activities in this stage include training, bug-fixing and maintenance. The costs at this stage are expensed.

|  |  |  |
| --- | --- | --- |
| **Comparison of Phases**  **Waterfall versus Agile Software Development Methods** | | |
| Methodology / Phase | Waterfall | Agile |
| Preliminary  (Expensed) | All customer requirements are documented, analyzed and determined technically feasible.  Business case has merit.  Financial resources are authorized by a company representative with budget authority. | High level customer requirements are determined technically feasible.  Business case has merit.  Financial resources are authorized by a company representative with budget authority. |
| Development  (Capitalized, except administrative overhead costs) | Design, coding and testing.  Customer is not engaged.  Projects are linear. | Design, coding, testing: requirements prioritized for the next iteration with feedback from the customer.  Customer highly engaged throughout.  Projects have multiple iterations, and each may re-prioritize the requirements. |
| Post-Implementation / Operation  (Expensed) | Customer accepts or rejects the project in the final approval stage. | Customer has accepted each iteration throughout the project, eliminating the risk of rejection in the final approval stage. |

The impact of appropriately capitalizing software development expenditures can be significant, and has a number of important benefits. Conservative treatment of agile projects -expensing all costs- limits the use and extent of agile-developed software projects. Software development resources are expensive and often limited. Appropriate capital allocation of these resources is important to the competitive health of the company, and also ensures consistent reporting and capital allocation within management and across organizations for investors.

For more information about the Software Development Lifecycle, please review: [http://en.wikipedia.org/wiki/Systems\_development\_life-cycle#Complementary\_to\_SDLC](http://en.wikipedia.org/wiki/Systems_development_life-cycle" \l "Complementary_to_SDLC" \t "_blank)

For more information about Waterfall SDLC please review methodology: [http://en.wikipedia.org/wiki/Waterfall\_model](http://en.wikipedia.org/wiki/Waterfall_model" \t "_blank)

For more information about Agile Methods and SDLC, please review: [http://en.wikipedia.org/wiki/Agile\_software\_development](http://en.wikipedia.org/wiki/Agile_software_development" \t "_blank)

For more information about the Agile Accounting, please contact: [agileaccountingstandards@agilealliance.org](mailto:agileaccountingstandards@agilealliance.org)