



Federal Accounting Standards Advisory Board

---

February 20, 2014

Memorandum

To: Members of the Board

From: Internal Use Software Working Group Co-Chairs:  
Curt Nusbaum, Transportation Security Administration  
Rebecca Shiller, National Security Agency

Through: Wendy M. Payne, Executive Director

Subj: Internal Use Software (IUS) Working Group, Progress to Date - **Tab C**

**OBJECTIVE:**

The objective of this session is to present to the Federal Accounting Standards Advisory Board (FASAB or, "Board") the progress and findings of the IUS Working Group and to seek Board input on the plan to pursue implementation guidance related to Statement of Federal Financial Accounting Standards 10, *Accounting for Internal Use Software* (SFFAS 10).

**BRIEFING MATERIAL:**

This transmittal memorandum includes an overview of the Working Group's activities, finding, and recommendations. Questions for the Board are included to solicit feedback from the Board on the planned approach and next steps. In addition to this Transmittal Letter, the Working Group has attached the following items:

**Attachment #1:** Mapping Sub-Group, Deliverable #1: Discussion of relevant requirements related to federal agencies' developed software and an assessment as to whether the Group believes such requirements are in alignment with current SFFAS 10 requirements.

**Attachment #2:** Standards Sub-Group, Deliverable #1: Discussion of current software development methods and an assessment on the challenges federal agencies face in complying with SFFAS 10 given the nuances of the various software development methods.

**Attachment #3:** Standards Sub-Group, Deliverable #2: Discussion on key FASAB Accounting Concepts related to the financial accounting of software under SFFAS 10.

**Attachment #4:** Summary matrix of issues and proposed solutions by development technique.

## **BACKGROUND:**

On 7 February 2013, the Transportation Security Administration (TSA) submitted comments to the FASAB Three-Year Plan (Plan) suggesting the Board include Accounting for IUS as a key topic in the Plan. The Board adopted TSA's suggestion and on 25 June 2013, the Internal Use Software Working Group held a kick-off meeting. During this meeting, the group split into three sub-groups; the Mapping Team, the Benchmarking team, and the Standards Team.

## **WORKING GROUP'S PRELIMINARY RECOMMENDATION:**

Based on the research performed by the IUS Working Group, the Group believes a Technical Release to SFFAS 10 would be appropriate. The Working Group would like to model the new Technical Release after *Technical Release 15: Implementation Guidance for General Property, Plant, and Equipment Cost Accumulation, Assignment and Allocation* (TR15). The Technical Release would provide greatly needed implementation guidance related to accounting for IUS and would focus on three main topics:

1. Framework and examples for distinguishing between development of major enhancements and minor enhancements and development to address bug fixes or keep software relevant, especially related to iterative software development models.
2. Clarification over the cutoff for capitalization that would broaden the acceptable criteria for capitalization cutoff by including not only final user acceptance testing, but also other indicators or transitioning events that are unique to a specific agency's policies.
3. Framework and examples of appropriate accounting for IUS when significant uncertainties relate to the development and operational use of a software product.

Additionally, although the application of Full Costing methodologies had not been a focus of the Working Group to date, a Technical Release that follows suit with TR15 would give the Board an opportunity to also provide Full Costing implementation guidance related to IUS in the same manner it provided guidance for general PP&E.

The working group considered additional approaches such as expensing all IUS costs and disclosing them within the notes to the financial statements, limiting the definition of items considered operations and maintenance, or capitalizing all costs until a system reaches full operating capability, but decided against these approaches. Some suggested approaches were not pursued because they cannot be supported with accounting theory. Other suggested approaches were not pursued because they likely result in replacing one existing problem with a new problem.

## **SUB-GROUP OBJECTIVES AND SUMMARY OF FINDINGS**

Objectives and a summary of findings for each group are included below.

**Mapping Team Objective:** Review and map existing Office of Management and Budget (OMB) software and information technology requirements and terms to SFFAS 10 to identify inconsistencies and omissions. Identify terminology and align the work that is required under all of the various standards where possible. The team reviewed four reporting requirements: the Clinger-Cohen Act of 1996, OMB Circular A-130 (*Management of Federal Information Resources*), OMB Exhibits 53 (*Agency IT Investment Portfolio*) and 300 (*Capital Asset Plan*), and Enterprise Architecture Documentation.

**Mapping Team Summary of Findings:** The Mapping Team's assessment of the ability of SFFAS 10 to meet the four reporting requirements reviewed revealed that SFFAS 10 does not provide the necessary information required by these documents. Specifically, the Mapping Team identified that:

1. Researched requirements focused heavily on reporting on budget execution and required actual dollars spent on IUS related activities in the year the expenditures occurred (i.e. in the budget year). This differs from SFFAS 10 which invokes the proprietary accounting concept of matching periodic amortization expenses to the use and economic benefit of an Agency's capitalized asset.
2. Researched requirements directed Agencies to report the full cost of their IUS activities as the total amount spent in a given budget year on IT investments, often broken out by development or maintenance. Conversely, SFFAS 10 defines full cost of IUS development as the total cost spent on a defined developmental period and could be presented at a value that includes accumulated expenditures over multiple years.
3. Documentation guidelines are more stringent within the reviewed requirements than those required by SFFAS 10.

The Mapping Group recommended that the larger group consider the following in moving forward with the final recommendation to the Board:

1. Reporting the full cost of IUS expenditures (outlays) in the year they were incurred to better align with budgetary principles. This could be achieved through a disclosure of total IUS expenditures in the Property, Plant and Equipment footnote under the Internally Developed Software section. This disclosure recommendation may be more appropriate for certain defined types of software development (refer to Standards Team deliverable 1).
2. Since certain types of software development cycles warrant the recognition of a capital asset, the Mapping team suggests that the group consider re-defining the IUS life-cycle to recognize only two phases (pre deployment and post deployment), where pre deployment (including conceptual formulation, analysis of alternatives, etc.) costs are capitalized and post deployment costs are expensed. This would better align to the existing general PP&E accounting standards and to the current legislative and budgetary reporting requirements. This would also reduce the costs of agencies having to differentiate between preliminary design and actual development.
3. The working group should also encourage FASAB to better define the documentation that agencies should retain in order to support accounting for IUS related to any changes adopted from above.

The Mapping Team believed that implementing these recommendations would better align SFFAS 10 with significant overseer requirements for IUS reporting. Specifically, implementing these recommendations would provide management and decision makers with budget data to aid in making future funding and investment decisions. Finally reporting actual expenditures would hold entities accountable to address significant variances from their budget formulation submissions.

*Reference Attachment #1: Mapping Group, Deliverable #1*

**Benchmarking Team Objective:** Research private industry and other CFO act agencies to identify best practices in analyzing and capitalizing IUS costs. Also review how the information is used by management (relevant and useful).

**Benchmarking Team Summary of Findings:** The Mapping Group concluded that private industry faces the same challenges as the private sector and have similar processes.

**Standards Team Objective:** Research current development cycles and identify challenges in applying SFFAS 10 as currently written. Also, identify accounting concepts applicable to accounting for software and, if appropriate, devise potential ways to affect change to the current standard that still align with the concepts.

**Standards Team Summary of Findings, Software Development Cycles:** The Standards Team found the software development model has dramatically changed since the issuance of SFFAS 10 in June 1998. The standard was written to conform to the linear/waterfall approach with three distinct life-cycle phases, which was the prevalent development approach at the time. While the standard acknowledges that various development frameworks exist, there is no incorporation of these differences in SFFAS 10. Thus, accounting for IUS becomes increasingly challenging as federal agencies move toward nonlinear models to develop software. Many of the issues in dealing with new development techniques and software architectures (such as the Cloud) are focused on timing of capitalization, costs of capitalization, and estimating useful life. Additionally, development of IUS as applied in SFFAS 10 is more focused on business/administrative types of applications. Agencies also have targeted use software that has a more focused scope supporting agency-specific mission needs. While the development costs may meet the threshold for capitalization, targeted-use software has many unknowns including deployment intentions and useful life. Any recommended modifications to the standard would need to clearly address the above issues and provide specific guidelines for applying the standard to the changing software development environment.

*Reference Attachment #2: Standards Team, Deliverable #1*

**Standards Team Summary of Findings, Accounting Concepts:** SFFAS 10 was designed around software life-cycle phases which include planning, development, and operations. The standard provides a framework for identifying software development phases and processes to help isolate the capitalization period (development phase) for internal use software (SFFAS 10, par 10.) Additionally SFFAS 10 focuses on the full cost (direct and indirect cost) incurred during the software development stage (SFFAS 10, par 16). However, the standard acknowledges that the life cycle management techniques that agencies can use may vary depending on the complexity and risk inherent in the project.

Currently, SFFAS 10 requires the reporting of all asset costs at the initial amount (i.e. amounts paid for them) and the cost of using them over each period is reflected through amortization. With modern software development models using an iterative approach, the current measurement techniques may not be accurately achieving the financial reporting objectives. For these types of development models in which it becomes increasingly costly to isolate development expenditures or in instances where it is

difficult to clearly define deployment and significant enhancements, it might be beneficial to consider a remeasurement model, such as value in use.

The concepts become the underlying basis for any modification in SFFAS 10. As documented, the concepts allow for several methods in approaching the measurement and recognition of the IUS asset created. Any accounting principle change must also be evaluated for consistency and appropriate disclosure guidance provided.

*Reference Attachment #3: Standards Groups, Deliverable #2*

The working group accumulated the issues and proposed solutions by accounting process and analyzed the alternatives based on development techniques (i.e. linear, cyclical, targeted-use).

*Reference Attachment #4: Summary Matrix*

#### **DETERMINATION OF NEXT STEPS:**

Q1: Does the Board oppose the Working Group moving forward in drafting a Technical Release to SFFAS 10 to provide implementation guidance related to the issues and challenges identified in the Working Group's Deliverables 1-3?

Q2: Assuming the Board does not oppose the planned way forward, does the Board have specific guidance or suggestions on the items the Working Group should focus on for inclusion in the draft Technical Release?

Q3: If the Board opposes the continuation of efforts to draft a Technical Release, does the Board have recommendations for an alternative path?

# Federal Accounting Standards Advisory Board

---

Internal Use Software Working Group

**Mapping Team Deliverable #1**

9/11/2013

Contents

Purpose ..... 3

Objective No. 1 - Identify relevant Federal legislation, regulation, or guidance related to a Federal Agency’s developed or purchased software reporting requirements..... 3

Objective No. 2 - Identify the purpose/intent of the software reporting requirements within the legislation, regulation, or guidance. .... 4

Objective No. 3 - Assess whether the financial reporting requirements of SFFAS No. 10 meets the relevant Federal legislation, regulation, or guidance related to a Federal Agency’s developed or purchased software reporting requirements. .... 7

*SFFAS No. 10, Accounting for Internal Use Software* ..... 7

*Comparison of Federal IUS Reporting Requirements to SFFAS No. 10*..... 8

*Summary* ..... 9

Objective No. 4 - Develop recommendations for the FASAB IUS Working Group to consider amending SFFAS No. 10 to achieve the relevant Federal legislation, regulation, or guidance related to a Federal Agency’s developed or purchased software reporting requirements..... 9

    Appendix A – References to Requirement Terms/Definitions..... 11

## ***Purpose***

The purpose of this paper is to present the Federal Accounting Standards Advisory Board (FASAB) Mapping Team Sub-Group's (hereafter referred to as the Mapping Team) research, conclusions, and recommendations on whether Statement of Federal Financial Accounting Standard (SFFAS) No. 10 – *Accounting for Internal Use Software (IUS)*, should be amended to achieve the reporting objectives of select Federal legislation, regulations, or guidance that impact IUS. To guide the Mapping Team's efforts, we developed the following objectives:

1. Identify relevant Federal legislation, regulations, or guidance related to a Federal Agency's developed or purchased software reporting requirements.
2. Identify the purpose/intent of the software reporting requirements within the legislation, regulation, or guidance.
3. Assess whether the financial reporting requirements of SFFAS No. 10 meets the relevant Federal legislation, regulation, or guidance related to a Federal Agency's developed or purchased software reporting requirements.
4. Develop recommendations for the FASAB IUS Working Group to consider amending SFFAS No. 10 to achieve the relevant Federal legislation, regulation, or guidance related to a Federal Agency's developed or purchased software reporting requirements.

The Mapping Team Sub-Group used the following references in performing our analysis:

- **Office of Management and Budget (OMB) Circular A-11:** *Preparation, Submission, and Execution of the Budget*
- **OMB Guidance on Exhibit 300:** *Planning, Budgeting, Acquisition, and Management of Information Technology Capital Assets*
- **OMB Guidance on Exhibits 53:** *Information Technology and E-Government*
- **FASAB – Statement of Federal Financial Accounting Standard No. 4 – Managerial Cost Accounting Concepts and Standards for the Federal Government** (SFFAS No. 4)
- **FASAB – Statement of Federal Financial Accounting Standard No. 10 – Accounting for Internal Use Software** (SFFAS No. 10)

## ***Objective No. 1 - Identify relevant Federal legislation, regulation, or guidance related to a Federal Agency's developed or purchased software reporting requirements.***

The Mapping Team Group reviewed various Federal legislation, regulations, and Federal guidance to assess whether there were specific reporting requirements related to a Federal Agency's developed or purchased software. Based on the Mapping Team's research, the following four items were selected as having the most relevance to a Federal Agency's requirements to report on developed or purchased software:

- [Clinger-Cohen Act \(CCA\) of 1996](#)
- [Office of Management and Budget \(OMB\) Circular 130](#)
- [Office of Management and Budget \(OMB\) Exhibit 53 and Exhibit 300](#)
- [Enterprise Architecture Documentation](#)



In the following section, the Mapping Team discussed each of the four items in detail.

***Objective No. 2 - Identify the purpose/intent of the software reporting requirements within the legislation, regulation, or guidance.***

The main purpose of this section is to discuss the Mapping Team’s assessment of the legislation, regulation, or guidance on developed or purchased software reporting requirements. Each of the four items will be discussed separately.

***Clinger-Cohen Act (CCA) of 1996*** – The CCA was enacted to improve the way the Federal Government Agencies acquire, use, and dispose of Information Technology (IT). Originally titled the Information Technology Management Reform Act (ITMRA) of 1996, CCA was part of the National Defense Authorization Act for FY1996. This act established a comprehensive approach for Executive Agencies to improve the acquisition and management of their resources by:

- Focusing information resource planning to support their strategic missions;
- Implementing a capital planning and investment control (CPIC) process that links to budget formulation and execution;<sup>1</sup>
- Rethinking and restructuring the way they do their work before investing in Information Systems; and
- The need to establish effective leadership. CCA requires each Agency to have a Chief Information Officer (CIO).

The CCA’s main objective is to integrate and streamline the CPIC process with budget, financial, and program management decisions<sup>1</sup>. CCA is program management and results measurement focused.

***Office of Management and Budget (OMB), Circular A-130*** - With respect to purchased or developed software, the purpose of OMB Circular A-130, *Management of Federal Information Resources*, is to provide OMB policy and guidance on IT management in the Federal government. OMB Circular A-130 (hereafter referred to as A-130) is focused on determining the capital planning needs of Federal Agencies and in establishing appropriate controls over IT investments.

Appendix IV to A-130 states, “The Clinger-Cohen Act ... grants to the Director of OMB various authorities for overseeing the acquisition, use, and disposal of IT by the Federal government, so as to improve the productivity, efficiency, and effectiveness of Federal programs.” Specific to IT management, A-130 includes specific guidelines that require Agencies to:

---

<sup>1</sup>Section 5122 requires each Agency to “provide the means for senior management personnel of the executive Agency to obtain timely information regarding the progress of an investment in an information system, including a system of milestones for measuring progress, on an independently verifiable basis, in terms of cost, capability of the system to meet specified requirements, timeliness, and quality.”

- Develop Agency-specific policies and procedures that provide for timely acquisition of required IT;
- Maintain an inventory of the Agency's major information systems;
- Use the guidance provided in OMB Circular A-11, Planning, Budgeting, and Acquisition of Fixed Assets, to promote effective and efficient capital planning within the organization;
- Document CPIC processes and provide to OMB consistent with the budget process;
- Document the Agency's Enterprise Architecture and provide to OMB as significant changes are incorporated; and
- Update the Agency IT Capital Plan twice annually and submit annually to OMB with the Agency budget submission.

In order to report actual costs and to use these costs going forward as valid estimates for budget formulation-execution comparisons (for use in the Agency IT Capital Planning), the full costs must be captured for the various IT development phases and recorded appropriately. A-130 defines full cost as all direct, indirect, and general and administrative costs incurred in the operation of an information processing service organization.

***Office of Management and Budget (OMB), Exhibit 53 and Exhibit 300*** – Every Agency subject to Executive Branch review is required to submit an annual IT investment portfolio. Included in the portfolio are Exhibits 53 and 300. The IT investment portfolio is integrated into each Agency's overall budget submission/performance budget justification to demonstrate adherence to overall programming and investment objectives.

OMB uses information recorded within these exhibits to ensure that IT investments align with and support each Agency's strategic plans and that Agencies are using a disciplined CPIC process to manage their IT initiatives. OMB bases budgetary resource decisions on information reported in the exhibits, makes sure that investments are in accordance with OMB policies, and reports to Congress whether IT investments are or are not properly being executed against outlined goals.

In requiring IT investment information to be reported in each Agency's IT investment portfolio, OMB is attempting increase IT investment visibility throughout the federal government. Increased visibility helps reduce the amount of funding wasted on duplicative IT investments, cost overruns, schedule slippage, etc. By presenting information reported in the Exhibits on the IT Dashboard, the Federal government is more effectively able to share information with the public, thereby increasing accountability. The ultimate goal of the IT investment portfolio and Exhibits 53 and 300 is to make investment information transparent to the U.S. public.

Exhibit 53 is a report of all Agency IT investments, including all major and non-major investments. Agencies report on all Federal budgetary resources used to fund IT investments, including prior year (PY) actual expenditures, estimated current year (CY) resources based on enactment, and estimated future budget year (BY) resources. Important investment elements for each investment are documented, including investment description, investment status, alternatives evaluations, etc.

Exhibit 53 also requires Agencies to report specific IT investment information related to many different types of development models, including investments in the cloud environment, investments for mission delivery, and investments in infrastructure.

Exhibit 300 is a Capital Asset Plan completed by each Agency for all major IT systems and IT budget initiatives. The Exhibit further describes the budget justification of capital asset investments and each investment's alignment with strategic and performance goals, including the investments outlined in Exhibit 53.

Exhibit 300 provides overall investment information and justification, outlines the investments alignment with program priorities, and provides a summary of costs for each investment. The summary of costs are presented as actual PY obligations, estimated CY obligations, and estimated BY obligation. This break out by fiscal year is similar to how the Exhibit 53 requires budgetary resources to be broken out. Exhibit 300, like Exhibit 53, requires cost information to be reported for development/modernization/enhancement expenditures and for operation and maintenance expenditures. Exhibit 300 recommends Agencies use a modular development model in the development of the software. However, the Exhibit is still required for more modern software models.

***Enterprise Architecture Documentation*** – The Enterprise Architecture Documentation (EAD) establishes a standardized approach to implementing and enabling Federal Enterprise Architecture. It provides for comparable architectures across the Federal Government that will be more useful in managing change and enabling mission success. EAD also promotes consistent and coherent understanding of program and service performance. It promotes inter-operability between programs, systems, and services – partnering between missions and Agencies, and optimizes inter-operability between programs against shrinking budgets. Finally, EAD is an authoritative reference for the design and the documentation of systems and services to primarily allow for verifiability of configuration. These purposes exist to encourage and promote a conceptual understanding and high-level declaration of an entity's current and potential future investments in IT.

The Enterprise Architecture then structures the documented response of each Agency into four Primary Outcomes: Service Delivery, Functional Integration, Resource Optimization, and Authoritative Reference. These exist to declare the eventual outcome and purpose of the investment concept. Following the identification of outcomes, an Agency must document eight Levels of Scope: 1. International, 2. National, 3. Federal, 4. Sector, 5. Agency, 6. Segment, 7. System, or 8. Application. These levels of scope help note how broad or narrow, at certain levels, the concept will be responsible for serving. Next, Agencies must disclose eight Basic Elements in the following areas: Governance, Principles, Method, Tools, Standards, Use, Reporting, and Audit and agencies must identify the six areas of Documentation: Strategic planning, business services, data and information, enabling applications, host infrastructure, and security. This forces Agencies to consider how they will structure, govern, develop, and maintain the concepts they have developed and recorded for review. The required six reference models: Performance, Business, Data, Application, Infrastructure, and Security support the analysis and reporting across Agencies. Lastly, the Enterprise Architecture provides four Plans and Views: Enterprise Roadmap, Transition Plan, Current Views, and Future Views that provide for phasing of development and execution as well as current and forward referencing.

The Common Approach to Federal Enterprise Architecture (CAFEA) accelerates Agency business transformation and new technology enablement by providing standardization, design principles,

scalability, an enterprise roadmap, and a repeatable architecture project method that is more agile and useful and will produce more authoritative information for intra- and inter-Agency planning, decision-making, and management.

***Objective No. 3 - Assess whether the financial reporting requirements of SFFAS No. 10 meets the relevant Federal legislation, regulation, or guidance related to a Federal Agency's developed or purchased software reporting requirements.***

In this section, the Mapping Team documents our assessment of whether SFFAS No. 10 achieves the IUS reporting requirements of the four items described in Objective No. 2. To complete this assessment, the Mapping Team first analyzed the objectives and reporting requirement of SFFAS No. 10 as described below.

***SFFAS No. 10, Accounting for Internal Use Software***

SFFAS No. 10 requires the capitalization of the cost of internal use software whether it is commercial “off-the-shelf” software (COTS), contractor-developed, or internally developed. Such software serves the same purposes as other general PP&E and functions as a long-lived operating asset. This standard provides guidance regarding the cost elements to capitalize, the timing and thresholds of capitalization, amortization periods, accounting for impairment, and other guidance.

This statement provides accounting standards for internal use software used by federal entities. Federal entities purchase COTS, hire contractors to develop substantially all of the desired software (contractor developed), or develop software internally using their own employees, with or without a contractor’s assistance (internally developed).

The scope of this statement is as follows:

- Software used to operate an entity’s programs (e.g., financial and administrative software, including that used for project management),
- Software used to produce the entity’s goods and to provide services (e.g., air traffic control and loan servicing), and
- Software that is developed or obtained for internal use and subsequently provided to other federal entities with or without reimbursement.

Software development phases in SFFAS No. 10 include planning, development, and operations. SFFAS No. 10 provides a framework for identifying software development phases and processes to help isolate the capitalization period for internal use software the federal entity is developing. SFFAS No. 10 maintains that “provisions of th[e] statement need not be applied to immaterial items” and that materiality should be determined by each entity. Furthermore, IUS must meet the following criteria to be considered capital:

- Estimated useful life of 2 or more years,
- not intended for sale in ordinary business, and
- has been acquired/developed for use by the entity.

### ***Comparison of Federal IUS Reporting Requirements to SFFAS No. 10***

In this section, the Mapping Team assessed whether SFFAS No. 10 met the developed or purchased software reporting requirements of the four Federal legislation, regulations, or guidance related to a Federal Agency's developed or purchased software reporting requirements described in Objective No. 2. Each of the Mapping Team's conclusions on the four items is discussed individually.

***Clinger-Cohen Act*** - SFFAS No. 10 does not address the CCA's main objective which is to integrate and streamline the CPIC process to support budget, financial, and program management decisions. Specifically, SFFAS No. 10 divides the software development process in three most common software development phases, but does not link these phases to an Agency's capital investment and planning model as described in the CCA.

***OMB Circular A-130*** - SFFAS No. 10 does not address Circular A-130 requirements to report on all costs incurred to implement an IT and/or software project. According to A-130, the development of a valid estimate for IT capital planning and budget formulation requires the full cost of an Agency's IUS development project to be captured, including direct, indirect, and general and administrative costs incurred. SFFAS No. 10 requires the capture of full cost (e.g., direct and indirect) only during the development phase, requiring that costs incurred in the Preliminary Design and Operational Phases be expensed in the year they were incurred. OMB A-130's primary objective is to address IUS expenditures and not the matching of expenses with the amortization of an asset's useful life as required by SFFAS No. 10.

***OMB 53 and 300*** - SFFAS No. 10 does not address the IUS reporting requirements of OMB Exhibits 53 and 300; specifically, the reporting of software development and budget execution by fiscal year, to include, actual dollars spent in previous years and budgeted amounts for current and future years. SFFAS No. 10 defers the recognition of expense until the project is completed and placed into service and the capitalized cost is amortized over the useful life of the software.

Relevant to SFFAS No. 10 is the *Development/Modernization/Enhancement Expenditures* investment element reported within Exhibit 53. These costs reflect the amounts spent or planned to be spent on developing a new IT asset or amounts spent to significantly modernize or enhance existing IT assets (i.e. costs deemed capital under SFFAS No. 10). Agencies are required to report steady state costs for each IT investment (i.e. operational costs under SFFAS No. 10). Although OMB Exhibits 53 and 300 report costs in a similar manner to the phases established in SFFAS No. 10, there are still many differences in presentation of information in the Exhibits when compared to information required under the standard. OMB Exhibits 53 and 300 are reported by fiscal year, to include actual dollars spent in previous years and budget amounts for current and future years. Conversely, under SFFAS No. 10, software costs could be reported at values that may span multiple fiscal years.

Additionally, SFFAS No. 10 requires that only development costs incurred to build an asset are recognized and carried on the financial statements at net book value. Future benefit received from the asset is recognized through periodic recognition of amortization expense in future reporting periods. The value reported in an Agency's balance sheet for a software asset in any given year would not coincide with the value of steady state costs reported in that year on the Exhibit 53, as the NBV is the total asset's cost less accumulated amortization, not the total amount spent.

Therefore, the reporting of actual IT costs by asset life-cycle phases under SFFAS No. 10 will not align with budget resources documented by fiscal year in Exhibits 53 and 300.

SFFAS No. 4 also requires assets to be reported at full cost; that is the total of direct and indirect costs incurred to develop or create the asset. OMB guidance on the preparation and submission of Exhibits 53 and 300 do not address or take into consideration indirect costs. Finally, SFFAS No. 10 only creates a distinction between integrated hardware/software systems and Internally Developed Software. OMB guidance on Exhibits 53 and 300 however acknowledges that there are various models and platforms related to IT investments.

***Enterprise Architecture Documentation*** - SFFAS No. 10 does not provide the information required by the Common Approach to Federal Enterprise Architecture described in EAD; specifically, data on the standardization, design principles, scalability, enterprise roadmap, or a repeatable architecture project method that is more agile and useful and for intra- and inter-Agency planning, decision-making, and management.

### ***Summary***

The Mapping Team's assessment of SFFAS No. 10's ability to meet the four Federal legislation, regulation, or guidance reporting requirements reviewed by the Mapping Team revealed that, SFFAS No. 10 does not provide the necessary information required by these documents. Specifically, the Mapping Team identified:

1. Legislation/regulation/guidance is heavily focused on reporting on budget execution, requiring the actual spend on IUS related activities in the year the expenditure occurred; this differs from SFFAS No. 10 which invokes proprietary accounting concepts to match the periodic amortization/depreciation expense to the use of an Agency's capitalized asset.
2. Legislation/regulation/guidance requires Agencies to report on the full cost of its IUS activities as the total amount spent in a given budget year on IT investment, often broken out by development or maintenance; whereas SFFAS No. 10 defines full cost of IUS development as the total cost spent on a defined development period accumulates and could be presented at a value that includes expenditures over multiple years.
3. Legislation/regulation/guidance documentation requirements are more stringent than those required by SFFAS No. 10.

***Objective No. 4 - Develop recommendations for the FASAB IUS Working Group to consider amending SFFAS No. 10 to achieve the relevant Federal legislation, regulation, or guidance related to a Federal Agency's developed or purchased software reporting requirements.***

As the working group moves forward in deciding if and how to amend SFFAS No. 10, the Mapping Team suggests, based on research of CCA, OMB A-130, Exhibits 53 and 300, and the EAD, that the following recommendations are considered during deliberation:

- Reporting the full cost of IUS expenditures (outlays) in the year they were incurred would align better with budgetary principles. This could be achieved through a disclosure of total

IUS expenditures in the Property, Plant and Equipment footnote under the Internally Developed Software section. This disclosure recommendation may be more appropriate for certain defined types of software development (refer to Standards Team deliverable 1).

- If certain types of software development cycles still warrant the recognition of a capital asset, the Mapping team suggests that the definition of an IUS life-cycle be re-defined to recognize only two phases (pre deployment and post deployment), where pre deployment (including conceptual formulation, analysis of alternatives, etc.) costs are capitalized and post deployment costs are disclosed. This would better align the existing general PP&E accounting standards with the current legislative and budgetary reporting requirements. This would also reduce the costs of agencies having to differentiate between preliminary design (i.e. the establishment of requirements) and actual development.
- We also suggest that in addition to amending SFFAS No. 10, that the working group encourage FASAB to better define the documentation that agency's should retain in order to support their accounting for IUS for any changes adopted from above.

The Mapping Team believes that implementing these recommendations will better align SFFAS No. 10 with the more significant legislation/regulation/guidance that impacts IUS reporting. Specifically, implementing these recommendations would provide management and decision makers with budget data to aid in making future funding/investment decisions. Finally reporting actual expenditure would hold entities accountable to address significant variances from their budget formulation submissions.

## **Appendix A – References to Requirement Terms/Definitions**

### **DEPARTMENT OF DEFENSE CHIEF INFORMATION OFFICER DESK REFERENCE. *Vol. 1:***

***Foundation Documents.*** Clinger-Cohen Act: Definitions and Acronyms of Commonly Used DoD CIO Terms. Aug. 2006. Sept. 9, 2013. Pg. 180.  
<<http://dodcio.defense.gov/Portals/0/Documents/ciodesrefvolone.pdf>>.

### **OFFICE OF MANAGEMENT AND BUDGET. *Circular No. A-130: Management of Federal***

***Information Resources.*** Section 6: Definitions. Feb. 8, 1996. Sept. 9, 2013.  
<[http://www.whitehouse.gov/omb/circulars\\_a130#6](http://www.whitehouse.gov/omb/circulars_a130#6)>.

### **OFFICE OF MANAGEMENT AND BUDGET. *Guidance on Exhibits 53 and 300.*** Section 4. What special terms should I know? 2012. Sept. 9, 2013.

<[http://www.whitehouse.gov/sites/default/files/omb/assets/egov\\_docs/fy14\\_guidance\\_on\\_exhibits\\_53\\_and\\_300.pdf](http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/fy14_guidance_on_exhibits_53_and_300.pdf)>.



FASAB IUS WORKING GROUP

# Software Development Cycles

---

## Deliverable 1

**Standards Team**

**10/28/2013**

A comparison of the application of current software development cycles and environments (i.e. cloud, shared services) to the Statement of Federal Financial Accounting Standards 10, Accounting for Internal Use Software (SFFAS 10).

## Contents

PURPOSE .....	3
TECHNICAL LITERATURE.....	3
SUMMARY .....	3
SOFTWARE DEVELOPMENT APPROACHES .....	3
<i>ISSUE #1: TIMING OF CAPITALIZATION</i> .....	5
<i>ISSUE #2: COSTS OF CAPITALIZATION</i> .....	6
<i>ISSUE #3: ENHANCEMENTS</i> .....	6
SOFTWARE ARCHITECTURES.....	7
SOFTWARE DEVELOPMENT USING THE CLOUD ENVIRONMENT.....	7
<i>ISSUE #1: TIMING OF CAPITALIZATION</i> .....	8
<i>ISSUE #2: COSTS OF CAPITALIZATION</i> .....	9
<i>ISSUE #3: ENHANCEMENTS</i> .....	10
SOFTWARE DEVELOPMENT IN A SHARED SERVICES ENVIRONMENT.....	11
<i>ISSUE #1: INADEQUATE RESOURCES</i> .....	12
<i>ISSUE #2: STAKEHOLDER BUY-IN/ORGANIZATIONAL CULTURE</i> .....	12
SOFTWARE DEVELOPED FOR TARGETED USE .....	13
<i>ISSUE #1: USEFUL LIFE</i> .....	14
<i>ISSUE #2: MULTIPLE VERSIONS</i> .....	14
<i>ISSUE #3: DEPLOYMENT</i> .....	15
CONCLUSION .....	15
APPENDIX 1: SOFTWARE DEVELOPMENT MODEL DIAGRAMS.....	17
ENDNOTES .....	20

## **PURPOSE**

To document various software development cycles and environments (i.e. cloud, shared services) used today and the complexities of applying the Statement of Federal Financial Accounting Standards 10, *Accounting for Internal Use Software* (SFFAS 10), as currently written.

## **TECHNICAL LITERATURE**

- Statement of Federal Financial Accounting Standards 10, *Accounting for Internal Use Software* (SFFAS 10)
- Statement of Federal Financial Accounting Standards 6, *Accounting for Property, Plant, and Equipment* SFFAS 6)
- Federal Financial Accounting and Auditing Technical Release 5: Implementation Guidance on Statement of Federal Financial Accounting Standards 10: *Accounting for Internal Use Software* (TR 5)
- Statement of Federal Financial Accounting Standards 35, *Estimating the Historical Cost of General Property, Plant, and Equipment: Amending Standards of Federal Financial Accounting Standards 6 and 23* (SFFAS 35)

## **SUMMARY**

In applying SFFAS 10 for internally developed software (IDS), to include software developed by contractors, many challenges exist due to rapid development in the software industry. SFFAS 10, which followed Financial Accounting Standards Board (FASB) Accounting Standards Codification 350-40, *Internal Use Software* (ASC 350-40), was issued by the Federal Accounting Standards Advisory Board (FASAB) in June 1998. For IDS, the standard was written to conform to a waterfall approach with three distinct life-cycle phases. This standard will need to be revised to reflect complex software development approaches, such as incremental, and iterative development; as well as new software architectures and environments such as the cloud environment and the shared services environment.

## **SOFTWARE DEVELOPMENT APPROACHES**

In accordance with SFFAS 10, *software life-cycle phases include planning, development, and operations* (SFFAS 10, p. 10) as compatible with the Office of Management and Budget (OMB) and the United States Government Accountability Office (GAO) guidance issued in 1998. While SFFAS 10 acknowledges that various iterations of development exist, it references that only two common phases/transition points exist for all information systems- the beginning of development and the end.

Over time the software industry has evolved and today new development frameworks and various cycles exist that impede the consistent application of SFFAS 10 as it is currently written making it harder to distinguish the two aforementioned common phases/ transition points (the beginning and the end). A summary of common software development techniques are described below with additional illustrations provided in **Appendix 1.**<sup>1</sup>

- **Linear Development**
  - **Waterfall Model** – also referred to as the linear-sequential life cycle model in which each phase must be completed fully before the next phase begins.
  - **V-model** - verification and validation model in which each phase must be completed fully before the next phase begins; however, testing of the product is planned in parallel with the corresponding phase of development
- **Incremental Development** – Additional functionality is implemented in each increment/release
  - **Modular Development Model** – requirements are divided into various builds with multiple development cycles making the life cycle a “multi-waterfall” cycle. The cycles are divided into smaller, more easily managed modules. Working versions are produced during the first module and subsequent releases adds function to the previous release.
  - **Rapid Application Development (RAD) Model** – incremental model in which components are developed in parallel as if they are mini projects. The mini projects are delivered and then assembled into a working prototype.
- **Iterative Development** – repeats the cycle of design, build, and test until the desired functionality is completed.
  - **Iterative Model**- development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model. Not all requirements are gathered up front for the entire life cycle.
  - **Agile Model** – small incremental releases with each release building on previous functionality. Working software is delivered frequently (weeks rather than months) and it assumes the end users needs are ever changing in a dynamic environment.
  - **Spiral Model** – four main phases exist in which the software project repeatedly passes through these phases in iterations:
    - *planning* - requirements are gathered;
    - *risk analysis* - risks are identified and alternative solutions are reviewed;
    - *engineering* - the software is produced; and,
    - *evaluation* - the customer evaluates the outputs.

With incremental and iterative development models the ability to deliver a working software product in a shorter time frame (typically iterations of one to eight weeks) provides for a more efficient and effective way of managing federal IT projects. Iterations are based on user stories and customer requirements and have the flexibility to change development direction based on shifting priorities. Both the OMB<sup>2</sup>, and the GAO<sup>3</sup> have issued guidance on implementing information technology (IT) development techniques that use methods such as agile.

The use of these models creates several challenges when applying the principles of SFFAS 10.

- **Issue #1:** Determining when the software should be capitalized using the SFFAS 10 definitions of development phases.
- **Issue #2:** Determining what costs should be capitalized.
- **Issue #3:** Determining what constitutes a major enhancement requiring a new capitalized asset.

### ***ISSUE #1: TIMING OF CAPITALIZATION***

SFFAS 10 presents three phases of software development that follow a linear approach to an IT project: the preliminary design phase, the software development phase, and the post-implementation/operational phase. Capitalized cost *should include the full cost (direct and indirect cost) incurred during the software development stage. (SFFAS 10, p. 9).* Costs incurred during the preliminary design phase and the operational phase would be expensed in the period incurred.

This is inconsistent with the incremental and iterative methods of software development in which the three phases are blurred and can occur at the same time. A software product may be delivered and tested in its initial form with minimal processing capability. This allows for the federal agency to begin using the software in a limited manner. For example, initial data conversion may occur in conjunction with development and capabilities will continue to be enhanced with each new release/development sprint. Additionally, preliminary design may be occurring based on user stories and requirements that were a result of interaction with the preliminary software release. Fixes, generally occurring in the post-implementation phase will also be occurring. The flexibility to move between the three phases of software development is in direct contrast with a linear approach to accounting for software development.

Additionally, Technical Release 5 explains that each agency should develop and document the agency's policies and procedures around determining the beginning and ending points of development (i.e. the period in which capitalized costs are incurred) (p. 5). While the decision to pursue development exists, the completion of conceptual formulation, design, and testing of software alternatives does not coincide with the development for incremental and iterative models.

## **ISSUE #2: COSTS OF CAPITALIZATION**

Costs that should be capitalized in accordance with SFFAS 10 include *those for new software (e.g., salaries of programmers, systems analysts, project managers, and administrative personnel; associated employee benefits; outside consultants' fees; rent; and supplies) and documentation manuals.*

*For contractor-developed software, capitalized cost should include the amount paid to a contractor to design, program, install, and implement the software. Material internal cost incurred by the federal entity to implement the COTS or contractor-developed software and otherwise make it ready for use should be capitalized (SFFAS 10, p. 9).*

Examples of costs that should be expensed include the costs of data conversion, repair of minor design flaws, and minor upgrades. *Costs incurred after final acceptance testing has been successfully completed should be expensed (SFFAS 10, p. 9).*

Because the developers are concurrently working on fixes, data conversion, new development, and design for future enhancements, the costs associated with each are difficult to separate. Many project managements lump all software programmer time into development subjecting it to capitalization under the rules of SFFAS 10. In reality some of this time should be expensed. Additionally, while each increment/iteration requires approval to proceed, there is no final acceptance testing which would signal the move from capitalization to expensing of software costs. Lack of final acceptance testing many times makes it difficult to retain proper audit documentation to support the commencement of depreciation.

## **ISSUE #3: ENHANCEMENTS**

The rules of SFFAS 10 state that *the acquisition cost of enhancements to existing internal use software (and modules thereof) should be capitalized when it is more likely than not that they will result in significant additional capabilities. (SFFAS 10, p. 10)*

In incremental and iterative environments, each new increment/iteration is designed to provide additional capabilities in the software. Agencies have difficulty in establishing when the initial software product should be considered a completed working asset, and when continual development in a series of releases becomes significant enough to be considered a new software asset subject to capitalization.

SFFAS 10 guidance on amortization states that *for each module or component of a software project, amortization should begin when that module or component has been successfully tested. If the use of a module is dependent on completion of another module(s), the amortization of that module should begin when both that module and the other module(s) have successfully completed testing. (SFFAS 10, p. 12)*

With rapid release of working prototypes, there is difficulty in determining when amortization should begin. The user may be able to implement the software on a limited basis, but future development and capabilities are being continually added. To wait until

the final product is delivered at full capability is also inconsistent with the concept of an “in service” date.

## **SOFTWARE ARCHITECTURES**

“Software application architecture is the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. It involves a series of decisions based on a wide range of factors, and each of these decisions can have considerable impact on the quality, performance, maintainability, and overall success of the application.”<sup>4</sup>

Software architecture focuses on the environment in which the software is developed and used. The system, the user, and the business all are interrelated aspects of the software architecture. Development decisions are based on user needs, the IT infrastructure, and business goals. As agencies find ways to collaborate, share services, and rely on new types of infrastructure, the FASAB standards may need to be updated to reflect the changing environments.

## **SOFTWARE DEVELOPMENT USING THE CLOUD ENVIRONMENT**

In recent years there has been a general movement towards hosted software solutions that are flexible, scalable, internet-based, and typically purchased on a subscription basis, generally referred to as cloud services. These cloud services typically require minimal investment in actual software development and focus on delivering tailored solutions more rapidly than under the traditional waterfall approach. There are varying degrees of cloud services; however, the most common ones are: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

The level of software development costs associated with cloud solutions depends on the specific cloud model that is being implemented. Under the SaaS approach, the consumer does not manage or control the underlying cloud infrastructure (includes network, servers, operating systems, storage) or individual application capabilities, with the possible exception of limited user-specific application configuration settings<sup>5</sup>.

The capability provided to the consumer under PaaS is to deploy infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider onto the cloud. The consumer does not manage or control the underlying cloud infrastructure, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.<sup>6</sup>

The capability provided to the consumer under IaaS is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).<sup>7</sup>

### *Implications as a Cloud Service User*

Cloud service users primarily incur costs related to fees for service or fees for application software. However, other development costs are incurred based on different cloud models.

Of the three primary cloud models, the SaaS model generates limited, if any, development costs to be capitalized. SaaS models offer consumers flexibility in the configuration of application features activated, application interfaces, system integration, forms, and reports, but result in minimal development costs. However, if a service provider is charging a customer for the service of “software development”, the amount of these costs would need to be determined. Additionally, to provide more flexibility and customizable solutions, SaaS providers typically offer a module that enables the consumer to develop custom code that can be added on to the baseline solutions. Maintenance of custom code is typically the responsibility of the consumer, and could be considered software development costs. Under either the PaaS or IaaS models of cloud computing, the level of software development costs that would be considered for capitalization would be directly related to the specific applications that are created by the consumer and hosted on the provider’s infrastructure.

Due to the nature of cloud services, the use of the traditional waterfall method which requires extensive requirements gathering and analysis at the beginning of a project is difficult to apply. Cloud services provide the ability to deliver a working software product in a shorter time frame with limited development costs. Both the OMB, and the GAO have issued guidance on implementing flexible systems in shorter timeframes and more efficient cost savings.

The cloud architecture creates several challenges when applying the principles of SFFAS 10.

- **Issue #1:** Determining when the software should be capitalized using the SFFAS 10 definitions of development phases.
- **Issue #2:** Determining what costs should be capitalized.
- **Issue #3:** Determining how to allocate development costs to multiple projects concurrently
  - **IAAS**
  - **PAAS**
- **Issue #4:** Useful life determination
  - **How long does organization plan to retain hosted applications**
  - What would be useful life when IAAS or PAAS when using for multiple development efforts that may have varying useful lives

### ***ISSUE #1: TIMING OF CAPITALIZATION***

Software in a cloud environment is able to follow the traditional waterfall approach to development to a limited extent when the model is SaaS. Under this model, requirements for a system are developed, management authorizes and commits to a



project, and user acceptance testing occurs. However, the challenge arises when determining if the software meets the useful life test. The majority of SaaS models are paid on an annual subscription basis, but provide the opportunity for the consumer to cancel prior to each renewal. In the instances in which software is purchased as a subscription, the project is authorized and completed; however, it is indeterminable as to how long the software will be used to perform the intended function with a service life of 2 or more years.

Under both the PaaS and IaaS cloud solutions, the timing of capitalization becomes more challenging in that the platform and the infrastructure may host multiple applications. The underlying costs of the infrastructure and platform should be included in the capitalized cost of each application. However, allocating these costs and determining the timing may be difficult as applications with different useful lives are hosted on the infrastructure and platform.

Further, cloud subscriptions frequently must commence before the application can be configured to meet the consumer's needs or development can commence under IaaS and PaaS. Those configuration costs should be included in the capitalized costs, but the timing may not coincide with the actual go-live date of the application. In these circumstances, the standard does not provide clear direction as to when capitalization should begin. Organizations must determine whether capitalization should coincide with the commencement of the subscription, the completion of configuration, completion of acceptance testing, or the actual go-live date, all of which could result in a different timing of capitalization.

## ***ISSUE #2: COSTS OF CAPITALIZATION***

SFFAS 10 provides guidance for determining which costs should be capitalized and which costs should be expensed. However, this guidance is tied to the three linear phases of software development. Due to the nature of the costs within a cloud environment, determining which costs, if any, should be capitalized is a challenge. The major components of cloud solutions are the licensing fees for the subscription and the data storage fees. In order to meet capitalization criteria, the subscription period and the costs of the licenses should be evaluated in accordance with lease criteria as discussed in TR 5. Cost determination is further complicated by data and storage fees that may be charged on a monthly basis. Although those costs are typically expensed, if they relate to the hosting of the specific application and expect to be incurred throughout the useful life of the application, they may need to be evaluated for capitalizations.

Aside from the licensing fees, the nature of the SaaS model specifically does not provide the opportunity for development as the application is built and maintained by the solution provider versus the consumer. Therefore, upgrade costs are typically not incurred under the SaaS model of cloud solutions. However, costs associated with customized add-ons to the baseline software may result in enhancement costs if significant charges are needed.

Under PaaS and IaaS, the costs associated with development would be determined for each application and a portion of the infrastructure or platform costs should be included. However, the challenge is determining if those costs (whether they are general licensing, data, or storage costs), should be capitalized at all. Therefore, determining how those

costs should be allocated to the specific application being developed becomes a secondary challenge.

Additional considerations regarding the costs to be capitalized include asset disposition costs. The cost of general property, plant and equipment (including internal use software) includes installation and disposition costs. Under the cloud environment, consumers frequently incur costs to configure the IUS asset at startup and also to retrieve and download all of its data at contract expiration. The nature of these costs need be evaluated for capitalization as they relate to the overall asset cost. This approach is not congruent with the traditional waterfall approach.

### ***ISSUE #3: ENHANCEMENTS***

The rules of SFFAS 10 state that enhancements should be capitalized when they will result in significant additional capabilities.

In the SaaS cloud environment, each new iteration of software is developed and installed by the service provider as part of the licensing and maintenance agreement of the baseline software. Therefore, no enhancement costs would be expected for the baseline as they would be expensed. However, maintenance for upgrades built by the consumer on the baseline software are usually the responsibility of the consumer. Most service providers provide these services for a fee which should then be considered for capitalization.

In the PaaS and IaaS, development costs for enhancements would be capitalized under SFFAS 10 guidance if they meet the capitalization criteria. However, depending on the development approach used for PaaS and IaaS, the applicability of the waterfall approach to capturing costs would need to be determined.

### ***ISSUE #4: USEFUL LIFE***

SFFAS 10 defines useful life as a capitalized software project that has an estimated service life of 2 years or more.

Useful life is defined in SFFAS 6, as adapted from Kohler's Dictionary for Accountants, as *the normal operating life in terms of utility to the owner* (SFFAS 6, p. 17)

In the instances in which fully functioning software is hosted under the SaaS model, the project is authorized and completed; however, it is indeterminable how long the software will be used to perform the intended function since it is subscription based. If the subscription is based upon an annual renewal, the criteria for a service life of 2 or more years is not met under the existing standards, and the software would be expensed.

In the IaaS and PaaS, the challenge with determining useful has multiple components. If the term of subscription for the infrastructure or platform is on renewed annually, it would not meet capitalization criteria based on less than two year life. However, if the licensing agreements for the platform or infrastructure services themselves exceed the two year useful life, then they should be considered for capitalization.<sup>8</sup> Depending on the frequency of development and deployment under IaaS and PaaS, the useful life may be a combination of the baseline licensing agreement for the services, and expected level

of software development and deployment under these models. Individual software applications would have their own useful lives, but must take into consideration the overall licensing terms of the underlying IaaS and PaaS agreements need. If the useful of a particular application is expected to extend beyond the baseline IaaS and PaaS agreements, then a determination of whether to extend the IaaS and PaaS useful lives would be required or an adjustment to the application useful life.

#### *Implications as a Cloud Service Provider*

Cloud service providers will typically charge service fees to use the cloud, but will also incur various development costs for items such as web site development, development or acquisition of software to be used by the customer, in addition to infrastructure and maintenance costs. Costs related to the development of the cloud are currently capitalized under SFFAS 10, as it is software used to provide a service.

In many instances similar issues exist, as discussed above, when applying SFFAS 10 in instances where the Agency is a cloud service provider. Additional challenges for cloud service provider include ownership of the software asset(s), specifically when written agreements, if they exist, are silent as to ownership rights. In some instances, funding for various components of the cloud have been appropriated to multiple agencies (i.e. Agency A is appropriated funding for the infrastructure while Agency B is appropriated funding for the platform development and various Agencies receive funding for applications to be available for use among multiple entities).

Additionally, for SaaS models, applications become available-for-use to various entities on the cloud. If the entity has a need for the application, additional development costs might be incurred for configuring the application for specific agency needs. Such a scenario involves the consideration of imputed cost for the use of the software (expensed) with actual costs for enhancements/modifications (capitalized).

## **SOFTWARE DEVELOPMENT IN A SHARED SERVICES ENVIRONMENT**

In a shared services environment, IT services are centralized for an agency or for multiple agencies. It is aimed at improving processes and reducing operating costs by leveraging shared platforms and service delivery. While not explicitly detailed in SFFAS No. 10, clear guidance is given in the CIO Council's Federal Shared Services Implementation Guide (drafted to assist agencies with carrying out the processes outlined in OMB's Federal IT Shared Services Strategy).

SFFAS No. 10 outlines the terms in which agencies are to capitalize software packages, though not specifically for shared services, it does cover bundled packages and bulk purchases:

*Federal entities may purchase software as part of a package of products and services (e.g., training, maintenance, data conversion, reengineering, site licenses and rights to future upgrades and enhancements). Federal entities should allocate the capitalizable and non-capitalizable cost of the package among individual elements on the basis of a reasonable estimate of their relative*

*fair values. Costs that are not susceptible to allocation between maintenance and relatively minor enhancements should be expensed. (SFFAS 10, p. 10)*

*Each federal entity should establish its own threshold as well as guidance on applying the threshold to bulk purchases of software programs (e.g., spreadsheets, word-processing programs, etc.) and to modules or components of a total software system. That guidance should consider whether period cost would be distorted or asset values understated by expensing the purchase of numerous copies of a software application or numerous components of a software system and, if so, provide that the collective cost should be capitalized. (SFFAS 10, p. 10)*

Applying the standard adds a measure of difficulty when implementing shared services because of the lack of clarity giving way to issues/concerns that the guidance does not cover.

Challenges that exist in developing software in a shared services environment are as follows:

- **Issue #1:** Ensuring adequate IT resources and infrastructure for the timely implementation of the shared services program. Lack of resources and/or substandard resources will hamper the launch of the agency's shared services program.
- **Issue #2:** Ensuring early buy-in from key stakeholders is fundamental to effective implementation of the shared services environment. The organizational culture resists the transformation to a new way of doing business.

### ***ISSUE #1: INADEQUATE RESOURCES***

When entering into a shared services environment program, it is imperative that the necessary equipment is available to launch. Without this crucial step, the success of the program does not exist. It is in the Preliminary Design phase that management defines and identifies the required software/IT resources. While SFFAS No. 10 may be silent on what software is required, it specifically treats preliminary design phase costs as expenses, with capitalization occurring after the conceptual formulation, design and testing of possible software alternatives. The challenge comes in identifying the IT resources and determining each of the agency's responsibility for ensuring the acquisition of sufficient IT resources for proper implementation.

### ***ISSUE #2: STAKEHOLDER BUY-IN/ORGANIZATIONAL CULTURE***

There needs to be a top-down, bottom-up understanding of the decision to move to a shared services environment. When OMB implemented its Federal IT Shared Services Strategy for all federal agencies to move to shared services, it became imperative that management executives and staff-level employees alike adhere to the policy. The CIO Council's Federal Shared Services Implementation Guide that details how to implement it was issued in April 2013. Communication and understanding is the key.

Management is needed to authorize and implement the agency process for participating in a shared services environment. The OMB strategy and guidance are new releases for shared services. There is no specific coverage on this topic in SFFAS No.10 which is of

concern as with any new initiative. However, some direction, the software development life-cycle is covered as follows:

*Software's life-cycle phases include planning, development, and operations. This standard provides a framework for identifying software development phases and processes to help isolate the capitalization period for internal use software that the federal entity is developing. (SFFAS No. 10, p. 6)*

Staff-level employees are critically needed as they perform the day-to-day work and have the technical skills and knowledge necessary for conveying software specifications. Employees are resistant to change fearing that the legacy system in hand is better than the transitioning system.

## SOFTWARE DEVELOPED FOR TARGETED USE

Certain software is developed for a specific targeted use of the entity, as opposed to business-type activities (i.e. an ERP or HR system). Software that is developed internally to meet the targeted use of the reporting unit (i.e. Agency, Department) does not typically follow the development cycle and usage as defined in SFAAS No. 10.

In accordance with SFFAS 10, *entities should capitalize the cost of software when such software meets the criteria for general property, plant, and equipment (PP&E)* (SFFAS 10, p. 15).

*General property, plant, and equipment is any property, plant, and equipment used in providing goods or services. General PP&E typically has one or more of the following characteristics:*

- *It could be used for alternative purposes (e.g. by other Federal programs, state or local governments, or non-governmental entities) but is used to produce goods or services, or to support the mission of the entity, or*
- *It is used in business-type activities, or*
- *It is used by entities in activities whose costs can be compared to those of other entities performing similar activities (e.g. Federal hospital services in comparison to other hospitals).* (SFFAS 6, p. 23)

*Internal use software is specifically identifiable, can have determinate lives of 2 years or more is not intended for sale in the ordinary course of operations, and has been acquired or constructed with the intention of being used by the entity* (SFFAS 10, p. 38)

Challenges that exist as they relate to targeted use software are as follows:

- **Issue #1:** Determining an appropriate useful life for software that is developed for target-specific needs.
- **Issue #2:** Distinguishing enhancements and an intended useful life for applications that are developed to perform a targeted need, in which numerous

versions need to be supported and deployment of the version is contingent on the specified target.

- **Issue #3:** Defining development activities for instances when targeted use software is deployed directly into operations before product testing is completed and formal user acceptance has occurred.

### ***ISSUE #1: USEFUL LIFE***

In accordance with the accounting standards, software should have an estimated service life of 2 years or more in its intended function and utility to the owner.

In the instances in which software is developed for target-specific needs, the project is authorized and completed; however, it is indeterminable as to whether the software will be used to perform the intended function with a service life of 2 or more years. This software may never be deployed; may be deployed for a single target and used until the target-specific mission is completed (hours to years); or, may be deployed for one target and then shelved for use on a different target in the future that is lagging on cyber-technologies (i.e. third world nation).

### ***ISSUE #2: MULTIPLE VERSIONS***

When determining whether a capitalizable enhancement exist, the accounting standard states that *enhancements normally require new software specifications and may require a change of all or part of the existing software specifications as well.* (SFFAS 10, p. 26)

Additionally, in determining the useful life, SFFAS 10 references that such useful life should be consistent with the planning of the software's acquisition:

*Software that is capitalized pursuant to this standard should be amortized in a systematic and rational manner over the estimated useful life of the software. The estimated useful life used for amortization should be consistent with that used for planning the software's acquisition* (SFFAS 10, p. 32).

Certain software is developed for targeted use purposes in which the development and maintenance of numerous versions/instances is necessary; and as such, the existing application is not subject to a "required upgrade" to the latest version. Targeted use applications need multiple instances due to varying platforms/operating systems (i.e. one software application to perform a target-specific need may need to be developed for Windows, Android, iOS, etc. Within the various platforms, application instances are tweaked for various releases (i.e. Windows 7, Windows Vista, Windows XP, etc). When a version of the operating system is upgraded, an enhancement to the software product will need to be made; however, the old instance is still in use and not impaired/obsolete because the targets do not necessarily upgrade as well. For most business applications that are deployed to a customer base, the newest versions of the software typically replaces or is intended to replace the previous version. In this model, the useful life for the enhancement (i.e. newer version) and the impairment of the older version would be evaluated.

For targeted use applications where multiple versions are maintained, "utility to the owner" (as discussed in Issue #1 above) is unknown as it is dependent on commercial industry trends and targets. Certain instances may never need to be utilized, while

others could last for a substantial period of time. In the instance where multiple versions of a software application need to be developed and maintained, enhancements remain necessary to support the application on various platforms and it does not require a change or impairment to the existing software specifications.

In instances where each version must be held indefinitely based on various target needs, management is unable to determine whether such version will be deployed and the basis for determining a useful life upon deployment. Funding for the application is not based on an instance-level and the old software does not get replaced; therefore it is not expensed at the deployment of the newer instance.

### ***ISSUE #3: DEPLOYMENT***

The rapid pace of technological advancements that are being made by our adversaries imbeds too much uncertainty as to whether software projects will have a useful life equal to or exceeding two years. In addition, to meet intelligence and information assurance needs in real-time requires a condensed software development life-cycle that prohibits a reliable, consistent, and cost effective determination of whether software projects are in the software development phase as defined by SFFAS 10, as software could cycle in-and-out of development based on the urgency of the targeted use and can be deployed in conjunction with testing.

SFFAS 10 speaks to commencing amortization as follows:

*For each module or component of a software project, amortization should begin when that module or component has been successfully tested. If the use of a module is dependent on the completion of another module(s), the amortization of that module should begin with both the module and the other module(s) have successfully completed testing (SFFAS 10, p. 33)*

In some instances, software deployed directly into operation does not always meet the capability of the end-user and therefore the project must either be abandoned or returned to developers for further design activities. Without the designation of successful testing and the ability to distinguish between software components for the software projects, it is difficult to determine when amortization should commence. The standard emphasizes *the need for a clear point for ending the developmental phase* (SFFAS 10, p. 41) to determine the commencement of amortization; however, if this point is not defined, the software could be held in a work-in-progress account indefinitely.

Also, many times user acceptance testwork is not completed due to the urgency of the target-specific task.

## **CONCLUSION**

Software development has dramatically changed since the issuance of SFFAS 10 in June 1998. The standard was written to conform to the waterfall approach with three

distinct life-cycle phases, which was the prevalent development approach at the time. While the standard acknowledges that various development frameworks exist, there is no incorporation of these differences in SFFAS 10. Thus accounting for IUS becomes increasingly challenging as federal agencies move toward nonlinear approaches to develop software. Many of the issues in dealing with new development techniques and software architectures (such as the Cloud) are focused on timing of capitalization, costs of capitalization, and estimating useful life.

Federal agencies are also moving toward more shared services agreements in an effort to make better use of limited resources. Because, SFFAS 10 has no specific coverage of this topic, there is lack of clarity in applying the concept, especially with regard to asset value and ownership.

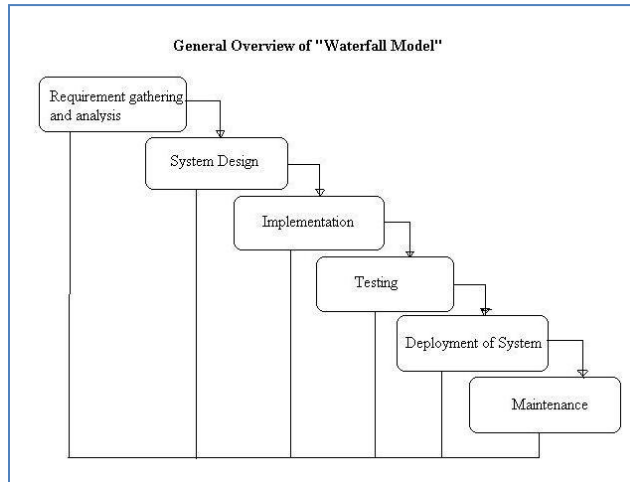
Finally, development of internal use software as applied in SFFAS 10 is more focused on administrative type applications. Many agencies have targeted use software that has a more focused scope. While the development costs may meet the threshold for capitalization, mission software has many unknowns including useful life.

Modifications to the standards will need to clearly address the above issues and provide specific guidelines for applying the standard to the changing software development environment.

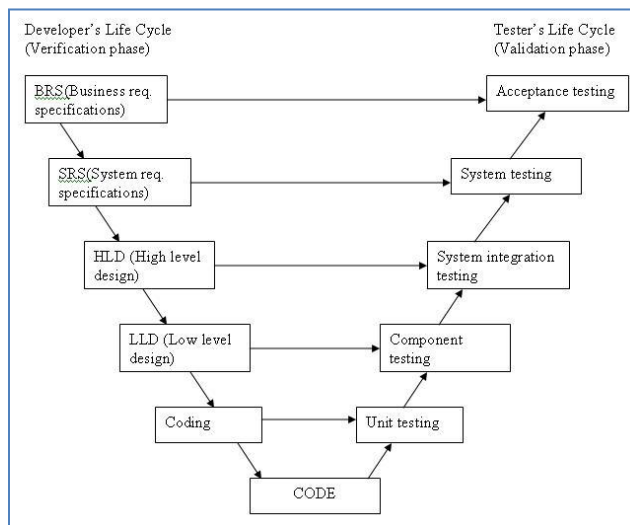


## APPENDIX 1: SOFTWARE DEVELOPMENT MODEL DIAGRAMS

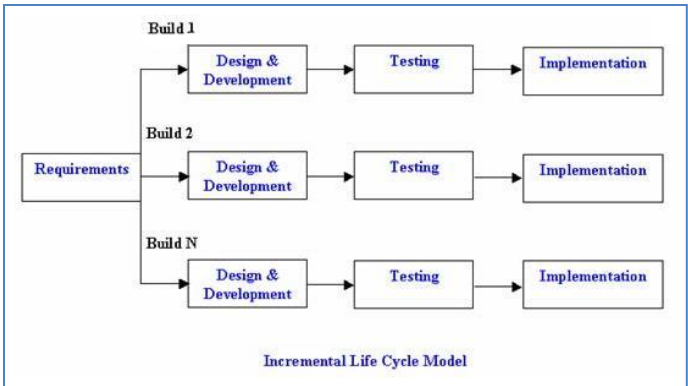
### *Waterfall Model*



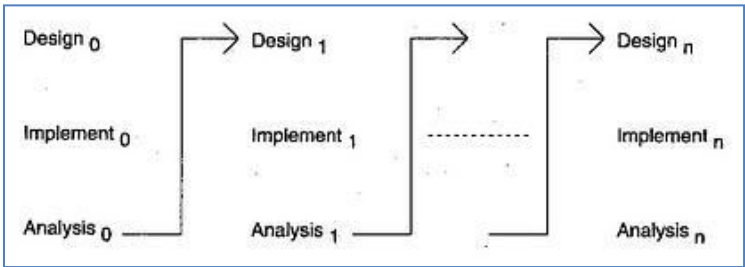
### *V Model*



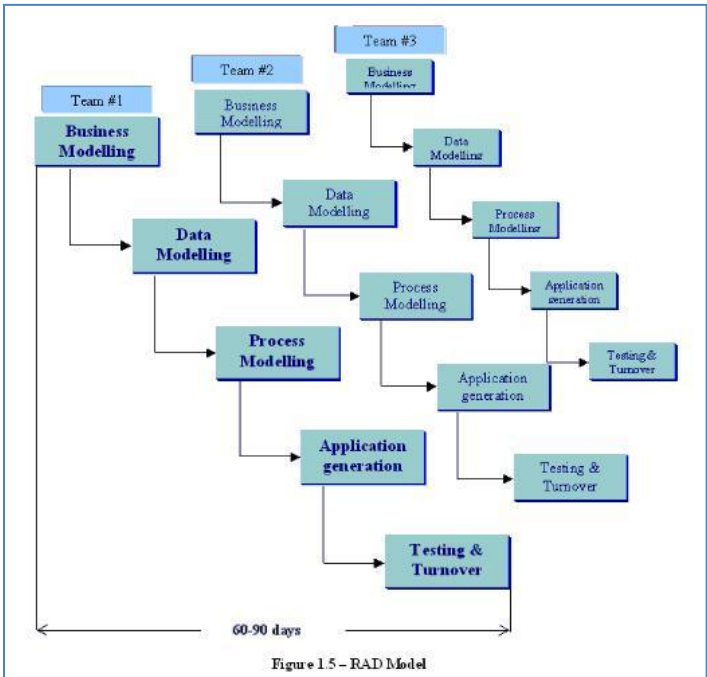
*Incremental Model*



*Iterative Model*



*RAD Model*



**Business modeling:** The information flow is identified between various business functions.

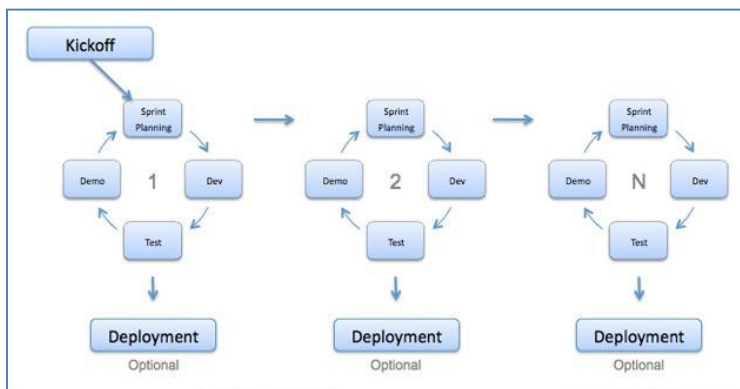
**Data modeling:** Information gathered from business modeling is used to define data objects that are needed for the business.

**Process modeling:** Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD of data objects.

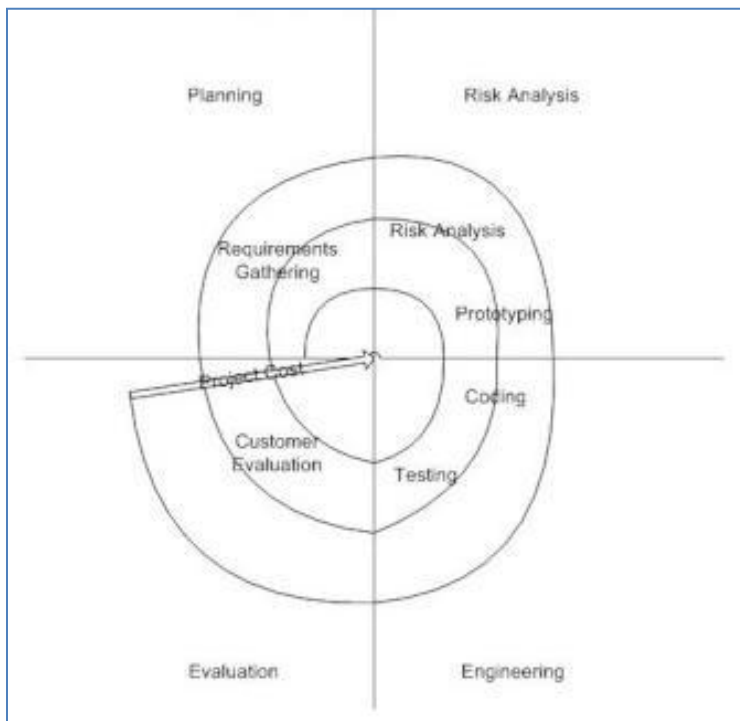
**Application generation:** Automated tools are used to convert process models into code and the actual system.

**Testing and turnover:** Test new components and all the interfaces.

### *Agile Development*



### *Spiral Model*



## **ENDNOTES**

---

<sup>1</sup> Information obtained and summarized from ISTQB (International Software Testing Qualification Board) Foundation Level Certification for Software Tester syllabus (Version 2011) and study material (<http://istqbexamcertification.com/what-are-the-software-development-models/>)

<sup>2</sup> OMB Circulars A-130 and A-11, as well as the Capital Programming Guide, include modular development and contracting approaches for capital assets.

<sup>3</sup> GAO 12-681, *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods*

<sup>4</sup> Microsoft Application Architecture Guide, 2<sup>nd</sup> Edition – October 2009

<sup>5</sup> National Institute of Standards and Technology, Special Publication 800-145, NIST Definition of Cloud Computing, Recommendations of the National Institute and Standards and Technology, Peter Mell, Timothy Grance, Page 2

<sup>6</sup> IBID, Page 2

<sup>7</sup> IBID, page 3

<sup>8</sup> NOTE: Further discussion on lease accounting, as it relates to software subscriptions, is being reviewed by another FASAB working group.

FASAB IUS WORKING GROUP

# Key Concepts

---

## Deliverable 2

Standards Team

**9/11/2013**

A review of key Statement of Federal Financial Accounting Concepts (SFFAC) as they relate to the criteria and capitalization of Internal Use Software (IUS).

## Contents

<b>Introduction</b> .....	3
<b>SFFAC 1, “Objectives of Financial Reporting”</b> .....	4
<i>Key excerpts from Concept Statement</i> .....	4
<i>Application of Concept to current accounting standards</i> .....	5
<b>SFFAC 5, “Definitions of Elements and Basic Recognition Criteria for Accrual-Basis Financial Statements”</b> .....	7
<i>Key excerpts from Concept Statement</i> .....	7
<i>Application of Concept to current accounting standards</i> .....	8
<b>SFFAC 7 – “Measurement of the Elements of Accrual-Basis Financial Statements in Periods after Initial Recording”</b> .....	9
<i>Key excerpts from Concept Statement</i> .....	9
<i>Application of Concept to current accounting standards</i> .....	12
<b>Summary</b> .....	14

## Introduction

The Federal Accounting Standards Advisory Board (FASAB) created accounting concepts to guide the formulation of federal accounting standards. According to FASAB:

*Each Statement of Federal Financial Accounting Concepts (SFFAC) is part of a series of concepts statements intended to set forth objectives and fundamentals on which financial accounting and reporting standards will be based. The objectives identify the goals and purposes of financial reporting. The fundamentals are the underlying concepts of financial accounting-concepts that guide the selection of transactions, events, and circumstances to be accounted for; their recognition and measurement; and the means of summarizing and communicating them to interested parties.*

*The Federal Accounting Standards Advisory Board's (FASAB or "the Board") conceptual framework enhances the consistency of standards and serves the public interest by providing structure and direction to federal financial accounting and reporting. (FASAB Handbook, Version 11, p. 6)*

Because these guide accurate financial reporting and interpretation of the standards, they should be referenced and incorporated into any modifications of a standard.

There are three financial accounting concepts that directly impact accounting for and capitalization of Internal Use Software (IUS). These include *SFFAC 1 - Objectives of Financial Reporting*; *SFFAC 5 - Definitions of Elements and Basic Recognition Criteria for Accrual-Basis Financial Statements*, and *SFFAC 7 – Measurement of the Elements of Accrual-Basis Financial Statements*.

When updating Statement of Federal Accounting Standard (SFFAS) No. 10 – *Accounting for Internal Use Software*, it is essential to adhere to the fundamental concepts presented which include determining the objectives for financial reporting, recognizing the criteria for an asset, and measuring the cost of the asset accurately.

## **SFFAC 1, “Objectives of Financial Reporting”.**

### ***Key excerpts from Concept Statement***

SFFAC 1 considers the users of government financial information, when determining the objectives of financial reporting, and classifies the users into four major groups: *citizen, Congress, executives, and program managers* (par. 75).

### ***Financial Reporting Objectives***

SFFAC 1 contains four main objectives of financial reporting:

- 1 Budgetary Integrity – fulfilling the government’s duty to be publically accountable for monies raised through taxes and other means and for their expenditure in accordance with the appropriations laws that establish the government’s budget for a particular fiscal year and related laws and regulations (par. 13).
  - The focus is on recording actual data from budget execution against appropriations made by Congress in using existing budgetary standards in accordance with legal authorization (par. 113).
  - The use of budgetary resources relates to information on the costs of program operations and whether information on the status of budgetary resources is consistent with other accounting information on assets and liabilities (par. 119).
- 2 Operating Performance – evaluating the service efforts, costs, and accomplishments of the reporting entity; the manner in which these efforts and accomplishments have been financed; and the management of the entity’s assets and liabilities (par. 14).
  - Federal financial reporting should provide information that helps the reader to determine the efficiency and effectiveness of the government management of its assets and liabilities (par. 130).
- 3 Stewardship – assessing the impact on the country of the government’s operations and investments for the period and how, as a result, the government’s and the nation’s financial conditions have changed and may change in the future (par. 15).
- 4 Systems and Control – understanding whether financial management systems and internal accounting and administrative controls are adequate (par. 17).
  - Transactions are executed in accordance with budgetary and financial laws and other regulations.
  - Assets are properly safeguarded (par. 146).



### *Consistency Principle for Financial Reporting*

Financial information must have the following basic characteristics: understandability, reliability, relevance, timeliness, consistency, and comparability (p. 156).

With an emphasis on consistency, the concept states the following:

Financial reports should be consistent over time; that is, once an accounting principle or reporting method is adopted, it should be used for all similar transactions and events unless there is good cause to change. The concept of consistency in financial reporting extends to many areas, such as valuation methods, basis of accounting, and determination of the financial reporting entity. If accounting principles have changed or if the financial reporting entity has changed, the nature and reason for the change, as well as the effect of the change, should be disclosed (par. 163).

### *Purpose of the matching principal in Federal Agencies*

Because government services are not usually provided in exchange for voluntary payments or fees, expenses cannot be matched against revenue to measure “earnings” or “net income” as would be done in business accounting. Moreover, directly measuring the value added to society’s welfare by government actions is difficult. Nonetheless, expenses can be matched against the provision of services year by year. The resulting cost can then be analyzed in relationship to a variety of measures of the achievement of results (par. 124).

The accrual basis of accounting generally provides a better matching of costs to the production of goods and services, but its use and application for any given purpose must be carefully evaluated (par.197).

### *Application of Concept to current accounting standards*

When evaluating the criteria of determining the financial accounting for IUS, the operating performance objective is the most prevalent. SFFAS 10 distinguishes the costs by effort through defining phases (i.e. preliminary design, development and post-implementation) and amortizes the entity’s asset over a useful life which can be determined based on intended funding requirements. Assets are discussed in SFFAC 1 as follows:

In government, as in the private sector, assets are expected to provide benefits that outweigh costs... Expected benefits often are not cash inflows but rather are the services provided by the asset. Sometimes those services are provided to the government itself (e.g. government office buildings or motor pools). More often, the services are provided to the public (e.g. education and research development) (par. 65).

Additionally, SFFAS 10 emphasizes the need for full costing associated with a project. Full costing is addressed in SFFAC 1 as follows:

Full assignment of all costs of a period, including general and administrative expenses and all other indirect costs, is an important basis for measuring cost of service. However, full cost is not necessarily the relevant cost for making all decisions (par. 198).

If changes to the current accounting standards are made, it is important to consider the consistency characteristic and matching principal to ensure proper disclosure for the nature and reason for the change.

=====

## **SFFAC 5, “Definitions of Elements and Basic Recognition Criteria for Accrual-Basis Financial Statements”**

When proposing changes to the IUS Standard, it is necessary to ensure that the updated guidance in implementing the standard does not conflict with Concept 5’s definition of an asset, definition of an expense, basic recognition criteria for assets, and the reporting entity who reports the asset (pg. 1).

### ***Key excerpts from Concept Statement***

**Definition of an Asset** – a resource that embodies the economic benefits or services the federal government controls. Paragraphs 21 – 35 describe essential characteristics of an asset.

**Definition of an Expense** – outflow of or other decrease in assets, an increase in liabilities, or a combination of both that results in a decrease in the government’s net position during the reporting period. Paragraphs 52 – 56 describe characteristics of expenses and revenues.

The statement establishes two basic recognition criteria that an item must meet to be a candidate for recognition in the body of a financial statement:

1. “the item must meet the definition of an element” (i.e., asset, liability, net position, revenue, expense)
2. “the item must be measurable, meaning a monetary amount can be determined with reasonable certainty or is reasonably estimable.”

“Meeting the basic recognition criteria is a necessary but not sufficient condition for recognition. Additional considerations for a recognition decision are measurement of the candidate for recognition and assessments of the materiality and benefit versus cost of the amount measured. Measurement entails selection of an appropriate attribute, such as historical cost, fair value, or expected value, and application of a measurement method. Measurement may require the use of estimates or approximations and, for items that meet the definition of an asset or a liability, an assessment of the probability that future inflows or outflows of economic benefits or services will result from the item” (pg. 2).

**Reporting Entity** - Questions related to which component entity of the federal government should report an item – review of authorizing legislation, appropriations act and related federal laws could establish the entity which has responsibility, based on the component entity that is responsible and accountable for “receiving, controlling, managing, and utilizing government assets... When no component entity has a

comprehensive relationship, the assets and other elements involved should be reported by the component entity most responsible for managing them” (par 12-13).

The economic benefits or services embodied in resources may be shared by the government and another entity (defined in paragraph 16 as entities external to the federal government or for a component entity this also includes other component entities) through specific arrangements. For example, the government and another entity may enter into a joint venture to share an interest in the resources committed to the joint venture. If so, each party may possess asset comprising its respective share of the benefits for services. Thus, both parties may have assets corresponding to their respective rights (par 28).

### ***Application of Concept to current accounting standards***

Modern software development models and government contracting guidance currently trend towards an iterative development model making it harder to distinguish the actual costs incurred (i.e. historical cost) associated with development, as defined in SFFAS 10, in a cost effective manner. While the software assets are currently valued at historical costs, as discussed in further detail below, the concept statements allow for other measurement approaches to be used; however, consideration for consistency of the valuation methodology should be contemplated.

The scope of SFFAS 10 includes “software that is used to produce the entity’s goods and to provide services (e.g. air traffic control and loan servicing)” as well as “software that is developed or obtained for internal use and subsequently provided to other federal entities with or without reimbursement.” With current government initiatives focused on shared services, in some instances arrangements exist in which software is developed, funded and managed by multiple reporting entities, particularly in the cloud environment. The financial reporting and accounting standards associated with software developed and managed by multiple entities currently do not address this issue.

=====

## **SFFAC 7 – “Measurement of the Elements of Accrual-Basis Financial Statements in Periods after Initial Recording”**

Updated guidance to SFFAS 10 for IUS, should not conflict with the qualitative characteristics of information in the financial statements, specifically with respect to measurement approaches and attributes.

### ***Key excerpts from Concept Statement***

#### *Financial Reporting Objectives and Qualitative Characteristics (pg. 6, par. 6)*

“The qualitative characteristics of information in financial reports are:

- a. Relevance – The capacity of information to make a difference in a decision by helping users to form predictions about the outcomes of past, present, and future events or to confirm or correct prior expectations.
- b. Understandability – The quality of information that enables users to perceive its significance
- c. Reliability – The quality of information that assures that information is reasonably free from error and bias and faithfully represents what it purports to represent
- d. Comparability – The quality of information that enables users to identify similarities in and differences between two sets of economic phenomena
- e. Consistency – Conformity from period to period with unchanging policies and procedures
- f. Timeliness – Having information available to a decision maker before it loses its capacity to influence decisions”

**Measurement** – The act or process of assigning dollar amounts to the elements of the financial statements (asset, liabilities, etc) (pg. 1). Measurement of assets and liabilities is grouped into two broad areas: Measurement Approach, and Measurement Attribute and Method.

**Measurement Approach** – (pg. 7 par. 7a) attributes and methods used for measuring assets and liabilities affect how the information is reported and interpreted

- 1 “Initial Amount” – amount initially recorded (i.e. the historical cost or historical proceeds (which may be adjusted subsequently for depreciation/amortization/depletion
- 2 “Remeasured Amount” – an amount measured at each financial statement date, such as the fair value.

Note – the above approaches are further broken down into nominal and constant dollars in paragraph 12.

**Measurement Attribute and Method** – (pg. 7 par. 7b)

- 1 “Measurement Attribute” – measurable characteristic of an asset or liability such as fair value, or settlement amount
- 2 “Measurement Method” – varies depending on the attribute selected, for example if measurement approach selected is “remeasured amount” and measurement attribute is fair value, a measurement method could be researching stock market quotes or obtaining a professional appraisal.

Different measurement attributes and methods may be used for different assets and liabilities, and the selections made can affect the usefulness of reported information for decision making (pg. 7 par. 7b).

*Achieving the Financial Reporting Objectives (pg 13 and 14)*

Remeasured amounts of assets and liabilities are determined using one of several possible measurement attributes that reflect economic conditions at the financial statement date, including, for example, fair value or settlement amount. Remeasurement updates a previously determined carrying amount to reflect a change in the economic value of an asset or liability that has occurred since the previous financial statement date. A remeasured amount thus differs from an adjustment to an initial amount that does not reflect a change in value. For example, an increase in the accumulated depreciation balance on a building does not change the economic value of the building and does not constitute remeasurement of its carrying amount. Unless the value of the building itself is remeasured at, for example, its fair value, the reported amount will continue to be considered the initial amount. In contrast, an adjustment to an allowance for uncollectible accounts receivable due to an increased risk of noncollection constitutes remeasurement of the carrying amount, even when the gross amount of receivables is not remeasured, because the adjustment reflects a change in the economic value of the receivables—the anticipated net settlement amount (par. 20).

Assessment of which nominal-dollar approach – initial amounts or remeasured amounts – better enables achievement of one or more of the financial reporting objectives vary according to the kinds of information users need and the decision to be made. In practice, federal financial statements traditionally have followed a “mixed-attribute” model. That is, some assets and liabilities, such as general property, plant, and equipment, have been reported at initial amounts (adjusted for depreciation, depletion or amortization, if applicable), and others, such as direct loans and loan guarantees, have been reported at remeasured amounts. (par. 21).

Continuing to report assets and liabilities at their initially recorded amounts in periods following their acquisition or incurrence is a long established approach to financial reporting and users are accustomed to that approach. Initial amounts generally are reliable and objective, based on documented evidence, although subjectivity subsequently may be introduced through the assumptions or methods adopted for calculating depreciation or amortization, such as depreciable lives and salvage values, or as previously indicated, through the allocation of indirect costs (par. 23).

Proponents cite these advantages in support of reporting at their initial amounts the costs of inventory and capital assets and the resultant costs of providing programs and activities (referred to in the operating performance objective). These proponents believe that it is not useful to remeasure and report assets at their potential sales prices or settlement amounts when they are being held to provide services, rather than for sale. In this view, assets held to provide services should be reported at the amounts paid for them (or other initial amounts), and the reported cost of using them each period should be a function of that amount (par. 24).

Reporting remeasured amounts may introduce significant uncertainties and subjectivity into the information provided to users because of the extent of judgment involved in developing these estimates (par. 26).

Supporters of remeasurement believe that users require up-to-date information about the price of assets held for sale or to generate future cash inflows. Further they believe that users also need information about the cost of programs and other ongoing activities based on the current costs of the underlying assets, particularly infrastructure and other capital assets that likely were acquired many years ago (par. 27).

Similarly, supporters of remeasurement believe that remeasured amounts of assets and liabilities, especially for assets acquired many years ago, are more relevant than initial amounts for assessing an entity's current financial position, service potential, and ability to meet obligations when due, as well as the magnitude of the entity's current and probable future resource needs (par. 28).

If an entity reports initial amounts, the statement of net cost reports the expiring benefits from previously expended budgetary resources only when the underlying assets are consumed or sold. The statement of net cost does not provide information about changes that occurs in resource prices or the values of existing assets in the intervening periods. In contrast, if the entity reports remeasured amounts, the information reflects the capacity of the underlying assets to provide goods and services in changing circumstances. The statement of net cost captures the period-to-period changes in asset amounts (holding gains and losses) in the periods in which they occur and reports the resources consumed

at current amounts, information that can help users assess stewardship and operating results each period (par. 30).

### *Value in Use*

The measurement attributes discussed are those most commonly applied or available for use: *fair value*, *settlement amount*, *replacement cost*, *value in use*, and *fulfillment cost*. Additional measurement attributes may be developed in the future. Fair value and settlement amount may be used to determine either the initial amount (historical cost or historical proceeds) or the remeasured amount of an asset or liability. Replacement cost and value in use (for assets) and fulfillment cost (for liabilities) are not applicable for assessing initial amounts because they are attributes of assets and liabilities that an entity already has recorded. These attributes may be used to remeasure recorded amounts at subsequent financial dates (par. 36). *Value in use is the benefit to be obtained by an entity from the continuing use of an asset and from its disposal at the end of its useful life (par. 50).*

Value in use is a remeasured amount for assets used to provide services. It can be measured at the present value of future cash flows that the entity expects to derive from the asset, including cash flows from use of the asset and eventual disposition. Value in use is entity specific and differs from fair value. Fair value is intended to be an objective, market-based estimate of the exchange price of an asset between willing parties. Value in use is an entity's own estimation of the service potential of an asset that it holds to provide a specific service. Examples include inventory and equipment with a unique design and purpose, and special purpose buildings, such as prisons. In those cases, the value in use may be greater than the amount the entity could obtain from selling the asset because the selling price would need to accommodate the purchaser's need to adapt the asset to another purpose (par. 51).

The service potential of an asset may be difficult to assess when the asset is used in combination with other assets and the total assessment must be allocated to the individual assets. In those cases, the reliability, consistency, and understandability of the remeasured amounts may be lower than when a direct assessment can be made of the value in use of each asset. The relevance of value in use is high for assessments of an individual entity, both with regard to the entity's management and for users' evaluations of operating performance, especially the entity's efficiency and effectiveness in managing its assets. However, the entity-specific nature of value in use reduces inter-entity comparability (par. 52)

### ***Application of Concept to current accounting standards***

Currently, the standard reports all asset costs at the initial amount (i.e. amounts paid for them) and the cost of using them over each period is reflected through amortization. As discussed above, modern software development models are trending towards iterative development in which activities related to development are hard to isolate. As a result, tracking actual costs



using the initial measurement approach as the standard is currently written becomes increasingly costly. For these types of development models in which it becomes increasingly costly to isolate development expenditures or in instances where it is difficult to clearly define deployment and significant enhancements, it might be beneficial to consider a remeasurement model, such as value in use.

Utilizing a value in use model for development models in which the development period is not clearly defined or instances when software is developed for a special, unique purpose (i.e. mission-based software), could provide for the current cost of programs and reflect the capacity of the software to provide goods and services in changing circumstances (i.e. changes in technological advances). It would minimize the entity's judgment in defining a significant enhancement and allow for a methodology in which the total spend on the program (i.e. development and life cycle) is used to support the asset value and support the resources consumed at current amounts. However, it will increase the risk of subjectivity due to the extent of judgment involved to value the asset.

The value in use measurement model is based on the present value of future cash flows expected to be derived from the asset. A discount rate should be applied against the future cash flow projection to support the time value of money. Considerations in implementing the value in use remeasurement technique for software development should include the following:

- The reporting entity should be able to consistently distinguish when the software development costs cannot be clearly delineated and have proper documentation to reach their conclusion.
- Point of initial measurement for asset recognition should be established.
- It is imperative to have a determination of a value in use model reflecting the future value to support the asset value and changes in the statement of net costs. The model needs to include reasonable and supportable assumptions that represent management's best estimate of expected cash flows, including cash flows from use of the asset and eventual disposition over the remaining useful life of the asset. It can be based on the most recent financial budgets and should full costing (i.e. overhead that is directly attributed or allocated to the asset).
- Disclosure requirements should include the valuation process

=====

## Summary

*SFFAS 10: Accounting for Internal Use Software* was designed around a software's life-cycle phases which include planning, development, and operations. The standard provides a framework for identifying software development phases and processes to help isolate the capitalization period (development phase) for internal use software (SFFAS 10, par 10.) Additionally SFFAS 10 focuses on the full cost (direct and indirect cost) incurred during the software development stage (par 16). However, the standard acknowledges that the life cycle management techniques that agencies can use may vary depending on the complexity and risk inherent in the project (A-130, "Analysis of Key Sections," p. 63)

*SFFAC 1: Objectives of Financial Reporting* addresses full costing and recognizes that it is not necessarily the relevant cost for making all decisions. Accordingly, accounting systems should permit the calculation of the relevant costs needed for a range of decisions, as determined by the specific situation, and financial reports should reflect costs suitable to the purpose intended (par 198).

SFFAC 1 also focuses on the consistency principle, and the matching principle. If accounting principles have changed or if the financial reporting entity has changed, the nature and reason for the change, as well as the effect of the change, should be disclosed (par. 163).

A modification to SFFAS 10 would need to be evaluated in terms of disclosure requirements.

*SFFAC 5: Definitions of Elements and Basic Recognition Criteria for Accrual-Basis Financial Statements* is another key concept statement relating to the criteria and capitalization of IUS. In SFFAC 5 there are two basic recognition criteria for an item to be recognized in the financial statements – the item must meet the definition of an element (asset), and the item must be measurable (par 52-56). SFFAC 5 recognizes that additional considerations may be required when making recognition decisions. Measurement is a key element in determining what and how an item is to be presented on the financial statements.

Modern software development models and government contracting guidance currently trend towards an iterative development model making it harder to distinguish the actual costs incurred (i.e. historical cost) associated with development, as defined in SFFAS 10, in a cost effective manner. Alternative measurement methodologies would be evaluated against the measurement principles addressed in the concept statements.

Reporting Entity is also discussed in SFFAC 5. As government initiatives focus on shared services, with development and funding being managed by multiple reporting entities, determining the reporting entity becomes increasingly difficult. The current standard does not address this issue.

*SFFAC 7: Measurement of the Elements of Accrual Financial Statements in Periods after Initial Recording* focuses on measurement and measurement approach. Different measurement attributes and methods may be used for different assets and liabilities and the selections made can affect the usefulness of reported information (par. 7b). Thus, it is important to strive to achieve the Financial Reporting Objectives as stated in the concepts.

Currently, SFFAS 10 reports all asset costs at the initial amount (i.e. amounts paid for them) and the cost of using them over each period is reflected through amortization. With modern software development models using an iterative approach, the current measurement techniques may not be accurately achieving the financial reporting objectives. For these types of development models in which it becomes increasingly costly to isolate development expenditures or in instances where it is difficult to clearly define deployment and significant enhancements, it might be beneficial to consider a remeasurement model, such as value in use.

In conclusion, the concepts become the underlying basis for any modification in *SFFAS 10: Accounting for Internal Use Software*. As documented, the concepts allow for several methods in approaching the measurement and recognition of the IUS asset created. Any accounting principle change must also be evaluated for consistency and appropriate disclosure guidance provided.

Process	Reference in current standard	Development Technique	Issues	Proposed Solutions	Comments
Step 1: Begin Capitalization	SFFAS 10, Paragraph 16 Capitalized cost should include the full cost (direct and indirect) incurred during the software development stage. Such cost should be limited to cost incurred after: a) management authorizes and commits to a computer software project and believes that it is more likely than not that the project will be completed and the software will be used to perform the intended function with an estimated service life of 2 years or more and b. the completion of conceptual formulation, design, and testing of possible software project alternatives.	Linear/Sequential	None	NA	
		Cyclical	None	NA	
		Target	Targeted-use software does not follow formal acquisition strategies of an Agency as it is volatile based on current events and unknown if it is more likely or not that the project will be completed. Broad requirements are developed (i.e. track target) and then are refined over and over when software is pulled back after being placed into operation in a Beta form.	1. Expense targeted use software and disclose total investment in RSI.  2. Expense target use software. Better define Target-use software in a Technical release to support individual agency decisions to expense such software (i.e. because the definition does not meet the capital criteria set forth in SFFAS 10). No change to existing guidance; can be addressed and supported utilizing existing standards.	For solution 1, at what level? i.e. still need to track at individual level, only if exceeding threshold, all target-use effort, etc.?  For solution 2, an implementation guide can be issued with illustrations/examples of applying current standards to existing, common issues.
Step 2: Accumulate WIP - Determine costs to be capitalized	SFFAS 10, Paragraph 17 Such costs include those for new software (e.g. salaries of programmers, systems analysts, program managers, and administrative personnel; associated employee benefits; outside consultants' fees; rent; and supplies) and documentation materials	Linear/Sequential	Modular development could result in development of future modules and maintenance of deployed modules occurring simultaneously	1. Capture all costs incurred after management makes "go" decision through "go-live" decision. These costs may include planning for additional increments or maintaining, enhancing prior increments. Under the provisions of estimation techniques in SFFAS 35, meet periodically with the IT SMEs to determine an appropriate estimate of maintenance costs to exclude from the capitalized portion. No change to existing guidance; can be addressed and supported utilizing existing standards.  2. Define "application maintenance" or "operations & maintenance" in implementation guidance. Perhaps use the first sentence of the definition of "Operations" from OMB's FY14 Exhibit 53/300 guidance: "Day-to-day management of an asset i the production environment including, but not limited to, activities that operate data centers, help desks, operational centers, telecommunications centers, and end-user support services." Limiting the definition of maintenance to these items potentially provides support to agencies to capitalize all development costs, whether for enhancements or repairing bugs.  3. In a similar manner as Technical Release 15, Appendix B, provide illustrations of the way the agencies representing this working group (plus others) have successfully addressed this issue.	Is the thought here that cap costs equal "go" on module/spiral 1 through "go live" on the last module?
		Cyclical	Cyclical development results in developers performing multiple phases (planning, developing, testing, maintaining, enhancing) simultaneously. The ability to distinguish capitalizable vs. non-capitalizable activities is not cost-effective.	1. Capture all costs incurred after management makes "go" decision through "go-live" decision. These costs may include planning for additional increments or maintaining, enhancing prior increments. No change to existing guidance; can be addressed and supported utilizing existing standards.  2. Through a technical release, broaden the definition of "final user acceptance testing" in implementation guidance to determine instances when the capitalization period should commence.  3. In a similar manner as Technical Release 15, Appendix B, provide illustrations of the way the agencies representing this working group (plus others) have successfully addressed this issue.	
		Target	Targeted-use software typically follows cyclical development models, however great uncertainties over useful life, viability, and operational use are great.	1. Expense targeted use software and disclose total investment in RSI.  2. Expense target use software. Better define Target-use software in a Technical release to support individual agency decisions to expense such software (i.e. because the definition does not meet the capital criteria set forth in SFFAS 10). No change to existing guidance; can be addressed and supported utilizing existing standards.	

Process	Reference in current standard	Development Technique	Issues	Proposed Solutions	Comments
Step 3: Determine when placed in service	SFFAS 10, Paragraph 33 For each module or component of a software project, amortization should begin when that module or component has been successfully tested. If the use of a module is dependent on completion of another module(s), the amortization of that module should begin when both that module and the other module(s) have successfully completed testing.	Linear/Sequential	None	NA	
		Cyclical	Software is placed in service at IOC (initial operating capability). This could include only a small portion of the software's capabilities. Thereafter, additional deployments occur; however, the FOC (final operating capability) is not always defined.	1 Capitalize costs up to initial deployment, and disclose in RSI additional costs related to the software lifecycle (to include development and maintenance) thereafter.  2. Capitalize costs and begin amortizing at initial deployment. Continue to accumulate all costs until software is substantially complete. This might include maintenance and minor enhancements to prior deployments; however, costs are not easily distinguishable. Subsequent to substantial completion, annually disclose in RSI additional costs related to the software lifecycle thereafter and consider significant enhancements for capitalization. No change to existing guidance; can be addressed and supported utilizing existing standards.	
		Target	Software may not be placed into service or may be placed into service and taken out of service. Software may also be placed into service in its Beta form and then continued, rapid development will happen to modify the software specific to the changing target.	1. Expense targeted use software and disclose total investment in RSI.  2. Expense target use software. Better define Target-use software in a Technical release to support individual agency decisions to expense such software (i.e. because the definition does not meet the capital criteria set forth in SFFAS 10). No change to existing guidance; can be addressed and supported utilizing existing standards.	
Step 4: Determine additional capitalizable enhancements	SFFAS 10, Paragraph 25-27 The acquisition cost of enhancements to existing internal use software (and modules thereof) should be capitalized when it is more likely than not that they will result in significant additional capabilities. For example, in an instance where the federal entity adds a capability or function to existing software for making ad hoc queries, the cost would be capitalized. Enhancements normally require new software specifications and may require a change of all or part of the existing software specifications as well. The cost of minor enhancements resulting from ongoing systems maintenance should be expensed in the period incurred. Also, the purchase of enhanced versions of software for a nominal charge are properly expensed in the period incurred. Costs incurred solely to repair a design flaw or to perform minor upgrades that may extend the useful life of the software without adding capabilities should be expensed.	Linear/Sequential	None	NA	
		Cyclical	1. Distinguishing an enhancement from part of the ongoing cyclical development. New releases typically include bug fixes, minor upgrades, user-enhancements, as well as changes to functionality.  2. Enhancements continue for a number of years (typically beyond the initial useful life given to the software).	1. Disclose (in RSI or footnote) total amount spent related to software lifecycle in current year. Provides information to the user without having to distinguish enhancement type.  2. Capitalize all program costs from "go" decision through program completion.  3. Follow the objective of the contract; Language in FASB ASC 350-40-25-11 refers to external costs (i.e. contracts) related to specific upgrades and enhancements: If maintenance is combined with specified upgrades and enhancements in a single contract, the cost shall be allocated and the maintenance costs shall be expensed over the contract period. However, external costs related to maintenance, unspecified upgrades and enhancements, and costs under agreements that combine the costs of maintenance and unspecified upgrades and enhancements shall be recognized in expense over the contract period on a straight-line basis unless another systematic and rational basis is more representative of the services received. No change to existing guidance; can be addressed and supported utilizing existing standards.  4. Consider significant enhancements that extend the useful life and provide significant additional capabilities / functionality as capitalizable. Or, consider assigning longer useful lives to software assets when it is expected that development will be necessary to keep the asset up-to-date and performing its intended function. No change to existing guidance; can be addressed and supported utilizing	For solution 1, disclosure could be annual expenditures for CY plus four prior years or could be comparative broken out by software program and further by O&M vs. DME.  For solution 2, need to determine if costs would remain in WIP through project completion or if additional asset value would be added annually. What would be the related impact to depreciation?
		Target	Target-use software typically follows cyclical development model. Software is deployed and enhancements are made frequently to meet target needs.	1. Expense targeted use software and disclose total investment in RSI.  2. Expense target use software. Better define Target-use software in a Technical release to support individual agency decisions to expense such software (i.e. because the definition does not meet the capital criteria set forth in SFFAS 10). No change to existing guidance; can be addressed and supported utilizing existing standards.	
Step 5: Determine non-capitalizable software related costs	SFFAS 10, Paragraph 20 Costs incurred after final acceptance testing has been successfully completed should be expensed. Where the software is to be installed at multiple sites, capitalization should cease at each site after testing is complete at that site.	Linear/Sequential	None	NA	
		Cyclical	Final acceptance testing does not always occur.	1. Identify a "substantially" complete point (see definitions below) and then disclose in RSI amount spent related to software lifecycle in currently year after that point occurs.	
		Target	Final acceptance testing does not occur. The beta software is deployed into operations to see if it works. Then it is pulled back and additional development occurs based on user feedback and changing target specifications.	1. Expense targeted use software and disclose total investment in RSI.  2. Expense target use software. Better define Target-use software in a Technical release to support individual agency decisions to expense such software (i.e. because the definition does not meet the capital criteria set forth in SFFAS 10). No change to existing guidance; can be addressed and supported utilizing existing standards.	

**General note:** If RSI disclosure is made for total life-cycle of project, need to clearly define/distinguish types of costs that should be included in the disclosure; i.e. all costs related to the programs' life cycle (excludes costs of users and analysts).

### Definitions

Linear/Sequential Development Models	Each phase must be completed fully before the next phase begins. This can also include modular development where the phases and deployments are clearly delineated.
Cyclical Development Models	In cyclical development, the project goal is pursued in several short, successive consecutive cycles. Each cycle is relatively short and within each cycle, a portion of the project is carried out. Analysis, design, implementation and testing occur within each cycle. Cyclical development slices system functionality into increments (portions). In incremental development, different parts of the system are developed at various times or rates and are integrated based on their completion. In iterative development, teams plan to revisit parts of the system in order to revise and improve them.
Targeted-Use Software	Software developed in response to short-fused requirements to meet specific target-needs. Target-use software is customized to meet specific requirements and have some of the following characteristics: A) Operational Significance is unknown at the time of development B) It is unknown if the software will be ever be deployed or if deployed it is unknown as to the length of time (hours to years) to reach the target; C) Software is deployed directly into operations before testing is complete with no formal user-acceptance test work performed. D) Deployment of the software is dependent on the targets technology; therefore, multiple versions are developed and maintained to meet various technological stages. E) No alternative future use.
Life Cycle Support / Maintenance	The act of keeping software in a usable condition, including preventative maintenance, normal repairs, development to keep the software relevant, and other activities needed to preserve the asset so that it continues to achieve its intended capability.
Substantially Complete	Capture costs until the technology is available for its intended use. Deployed software meets a substantial portion of high-level and/or critical requirements and functionality.