

Chapter 11 - Computer Cost Allocation (Algorithm)

Authoritative Sources

Introduction

This section contains guidance for evaluating the accounting for computer programming and reprogramming costs and computer operating costs.

11-1 Allocation of Computer Operating Costs

11-1.1 General Principles

DCAA policy requires that where computer costs are material, the FAO audit staff should develop an understanding of computer cost composition and test the contractor's use of the criteria sufficiently to assure that costs are distributed in an equitable manner. If an algorithm is used, and costs distributed are significant, periodic audit evaluation of the algorithm is essential.

This coverage addresses a common situation where a contractor has a computer system designed to be responsive to only the internal needs of the organization. Adjustments will have to be made to the audit program to handle the other types of computer system environments which the auditor may encounter. Adjustments should be made on a case by case basis.

This section primarily addresses billing algorithms. However, many of our contractors distribute IT costs through general indirect cost allocations. In those cases auditors must still determine whether methods used to distribute IT costs are equitable. While algorithms based on resource utilization are generally preferable, an algorithm is not required if indirect cost distribution is equitable.

Cost Accounting Standard 418 as related to computer costs provides for consistent determination of direct and indirect costs. It provides criteria for the accumulation of indirect costs including service center and overhead costs in indirect cost pools and provides guidance on selection of allocation measures based on the beneficial or causal relationships between an indirect cost pool and cost objectives. Refer to CAS 418 (CAM 8-418) for additional details.

Billing algorithms used by contractors to allocate computer costs should be included in a contractor's disclosure statement in order for the disclosure statement to be considered adequate (see CAM 8-206).

11-1.2 Algorithm Development

A computer billing algorithm is a mathematical formula used to develop the amount to be charged a customer, contract or overhead pool for services. The formula is based

on such factors as type of equipment used, storage media utilization and space allocation, type of processing, response or turnaround time, and time of day services are provided. In a complex IT environment, a wide range of IT support is provided to various system users. Developing an algorithm to equitably distribute IT costs may incorporate all major IT resources or only a few. The greater the variation in types of application or services provided, the greater the need for a more complex algorithm. The cost of developing a complex algorithm, including subsequent recording of computer use through internal software, is normally compared with the benefit (exactness) of such an algorithm. If it can be demonstrated that an algorithm using only two or three resources is equitable, a complex algorithm is not necessary.

Resources typically measured and collected for construction of a user charge include:

- Central processor (CPU) time - the amount of CPU time required to accomplish a specific task.
- Computer memory requirements - many algorithms consider the amount of memory (bytes) used for each job.
- Input/output transactions - with the wide range of data input/output devices such as magnetic tape, disks, and terminals, algorithms often consider the number of times such equipment is accessed.
- Direct access storage requirements - tape and disk storage requirements are often considered, including the amount of disk workspace and number of tape devices and/or tape mounts required by each job.

Typically, accounting information is collected by operating system software for each user application. In addition, the operating system usually contains provisions for user-supplied routines to collect utilization data. Numerous software vendors have developed specialized software packages to reduce these data and generate a variety of management reports. Such packages often provide time-sequenced resource utilization statistics that can be used to develop billing criteria and make recommendations on improving overall system efficiency.

Billing information is usually generated by a billing algorithm. Often the final billing unit is an average resource unit incorporating the various algorithm components. A simple example is shown below:

$$\begin{aligned} \text{Resource unit} &= \text{CPU time} \times \text{coefficient} \\ &+ \text{Memory usage} \times \text{coefficient} \\ &+ \text{Input/output transactions} \times \text{coefficient} \\ &+ \text{Printer time} \times \text{coefficient} \end{aligned}$$

The coefficients, which include but are not limited to staff costs, programming costs, and hardware costs, should be evaluated for applicability. Most often, coefficients

reflect a ratio between the cost of a specific resource and the total availability of the resource (for example, cost of CPU divided by total available CPU seconds.)

11-1.3 Audit Objectives in Algorithm Evaluation

When evaluating computer billing algorithms, audit objectives include:

- a. Developing an overall understanding of allocation methods used.
- b. Verifying that algorithm components accurately represent resources used.
- c. Validating that there are sufficient controls to assure that billings are processed in an accurate and reliable manner.
- d. Determining whether all applicable costs are included in the development of the coefficients.
- e. Validating that the individual rates or coefficients are accurate and properly applied.
- f. Testing allocation criteria to assure that computer cost allocations are equitable.

11-1.4 Algorithm Review Techniques

For purposes of algorithm evaluation, a structured audit approach is suggested as outlined in the following subparagraphs. A billing algorithm summary checklist, as shown in Figure 10-1-1, is often useful to control necessary audit steps.

- a. Determine billing formula risk and materiality. If billing algorithms do not distribute a material amount of contract cost (direct and/or indirect), the need for a detailed algorithm review may be obviated.
- b. Request contractor support for the billing formula:
 - (1) Explanation of the algorithm. Generally the contractor should have documented the algorithm. Consideration should be given to any tests (benchmarks) performed to validate the algorithm.
 - (2) IT resources used in the algorithm. The contractor should be able to identify which resources have been included in the formula and the rationale, if applicable, for excluding major resources.
 - (3) Cost distributed during recent periods.
 - (4) Accounting treatment of variances. This is a critical area as the timing of variance adjustments and accounting treatment can significantly impact costs distributed to contracts.

(5) Current inventory of IT equipment. This will be valuable when determining whether all appropriate resources are included in the algorithm. In addition, it is essential for adequate equipment maintenance and control that the contractor have detailed visibility of IT resources.

c. Compare billed IT costs with actual:

(1) Are procedures established for equitable and timely treatment of identified variances?

(2) If there are significant recent variances, has the algorithm been adjusted for more accurate cost distribution?

(3) Does the contractor compare costs for periodic runs of the same job; for example, payroll? Are significant differences investigated?

(4) Does the contractor make periodic revisions to projected rates as a result of changes in estimated costs or usage of a component?

(5) Are discounted coefficients offered for off-hours usage?

(6) Has an evaluation been made of the contractor's previous projections of computer component rates by comparison of actual rates to projected rates? What are the reasons for significant variances such as unplanned usage or nonusage, or the increase or decrease in costs? If the contractor makes periodic reviews of projected rates, arrange to audit these reviews. If there have been significant variances due to volume differences, perhaps more frequent reviews should be recommended.

d. Verify major IT resources. Critical considerations for an algorithm are whether it is based on verifiable usage data, and whether resources used in the algorithm accurately represent services provided system users. Consider whether:

(1) All major resources are included in the algorithm.

(2) Resource usage is based on verifiable data.

(3) Resources are costed appropriately.

(4) Algorithm components are restricted to IT resources.

(5) Lease agreements for equipment have been considered.

(6) Equipment costs are properly determined for each grouping.

(7) The algorithm includes any unallowable costs, such as excessive rental charges for IT.

- e. Evaluate coefficients and other factors:
 - (1) Are coefficients based on verifiable data?
 - (2) If there are outside sales of IT services, are the services comparable to in-house applications and are they priced comparably to in-house IT support?
- f. Manually compute the billing formula for selected major Government projects:
 - (1) Can the algorithm be computed using verifiable data?
 - (2) Is the manual calculation reconcilable to the machine output?
 - (3) Can coefficients and factor utilization be accurately verified?
 - (4) Are comparisons of items such as the ratio of cost input to IT billings reasonable?

Billing Algorithm Summary Checklist

Audit Step	Working Papers Reference	Auditor	Date
1. Risk evaluation			
2. Contractor support			
a. Obtain explanation of algorithm			
b. List IT resources in algorithm			
c. List distributed IT costs by quarter			
d. Identify accounting treatment of variances			
e. Identify IT policies/procedures for cost treatment			
f. Obtain current inventory of all IT equipment			

3. Compare billed IT costs with actual			
a. Variance treatment			
b. Timing of adjustments			
c. Are fixed-price/commercial type variances substantial			
4. Verify IT inventory (consider sampling techniques)			
a. Purchase agreements			
b. Are major resources in algorithm?			
5. Evaluate coefficients and other factors – Are coefficients based on verifiable data?			
6. Manually compute billing formula for major Government projects			
a. Is it based on available/verifiable data?			
b. Is the manual calculation reconcilable to machine form?			
c. Can coefficients/factors be verified?			
d. Are parity checks such as contribution to cost comparable?			

11-1.5 Billing Algorithm Example

When internal measurements are used, billing rates are developed to allocate the cost of each major component on the basis of the component's usage. These billing rates are usually computed annually and are developed by dividing the estimated annual cost associated with each component by the estimated annual usage of the component. The billing may be made in one of two ways: (1) separate billing rate for each component or (2) a single overall rate which is applied to equivalent units of usage for each component.

Computer costs can be distributed equitably using a wide range of mathematical techniques. As previously discussed, it is important that a contractor clearly document methods used, and base cost allocations on verifiable cost and utilization data.

The example in Figure 10-1-2 includes a five-resource cost allocation. For illustration purposes, one resource-magnetic tape drives-is traced through a weighting factor (coefficient) adjustment and the rate calculation. Coefficients are not essential but are included in many algorithms. Accordingly, a typical coefficient is included in the example.

Billing Algorithm Example

1. Formula resource components are:

<u>Resource Allocated</u>	<u>Unit of Measure</u>	<u>Charge/Prime Shift</u>
CPU	CPU hours	\$300/hr
Memory	1024 work block hours	\$5/hr
Disk Channel Time	Channel hours	\$25/hr
Tape Channel Time	Channel hours	\$10/hr
7- and 9-track Tape Drives	Elapsed hours	\$5/hr

2. The coefficient is computed using the following algorithm:

$$CFW = \frac{\text{Cost } r}{T \text{ Cost}} \times \text{Total } r \times \% \text{ used}$$

CFW = computer weighting factor or coefficient to equalize billings.

Cost r = cost of resources being allocated

T Cost = total IT costs to be allocated

Total r = number of resource units available

% used = percent resources are used

3. If we want to illustrate the weighting factor for tape drive utilization, we can assume the following data was available in contractor records.

Cost r = \$12,500

T Cost = \$3,000,000

Total r = 16 tape drives

% used = 70%

4. Substitute into the algorithm:

$$CFW = \frac{12,500}{3,000,000} \times 16 \times 70\% = .046$$

5. After developing an application weighting factor (coefficient), a rate is normally developed for the resource. Again for illustration purposes.

$$\text{Rate} = \frac{\text{Cost r}}{\text{Max r hours}} \times \frac{1}{\text{CWF}}$$

6. If contractor records show:

Cost r = \$12,500

Max r (prime shift) 16 tape drives

X 40 hrs/wk X 52 wks = 33,280

(second shift) 16 tape drives

X 40 X 52 X 50% disc 16,640

49,920

7. Substituting:

$$\text{Rate} = \frac{\$12,500}{49,920} \times \frac{1}{.046} = \$.25 \times 21.7 = \$5.43$$

8. As shown above, manually calculating the rate for tape drives shows an actual rate of \$5.43. If a billing rate of \$5.00/hr is used and utilization forecasts are accurate, tape drive cost will be underabsorbed.

As billing algorithms vary widely, this example should not be viewed as typical. However, it does demonstrate potential algorithm complexity. Accordingly, the approach suggested in 7-102.4 provides a frame-work for developing an audit opinion without evaluating and testing each component of the algorithm. If each factor or algorithm component cannot be verified by historical or current data, risk that costs are unequitably distributed is greatly increased. In such cases, the audit report should recommend that billing algorithms be based on verifiable data and that they include major IT resources used.

In many instances contractors may simplify the billing process. The example below addresses CPU costs only (other resources would be billed similarly), and if estimated CPU utilization is reasonable, billed costs would be equitable.

$$\text{Coefficient } t = \frac{\text{Cost of CPU for billing period}}{\text{Available CPU seconds for billing period}}$$

Cost of CPU for billing period = \$15,000

Available CPU seconds = 720,000

$$\text{Coefficient } t = \frac{\$15,000}{720,000} = \$0.020833$$

Billed amount = \$.020833 X CPU seconds consumer for each job

11-2 Significant Nonrecurring Costs of Computer Programming and Reprogramming

11-2.1 General Principles

Equity in accounting for significant nonrecurring costs of computer programming and reprogramming usually requires that such costs be capitalized/amortized. The initial programming costs are incurred in order to place the computer into operation and as such are normally as much a part of the initial costs of the computer as are the equipment installation costs. A major change in either the equipment or the system usually involves the incurrence of significant reprogramming costs. These costs will normally benefit future periods in much the same manner as major modifications of the equipment. On the other hand, established programs are subject to minor refinements and improvements, the costs of which are chargeable to current operations in much the same manner as minor repairs.

11-2.2 Amount to Be Capitalized

The amount of programming or reprogramming costs to be capitalized should represent the actual costs incurred by the contractor in preparing and testing the program; that is, all applicable direct and indirect costs should be included up to the point the program becomes operational.

11-2.3 Amortization Period

The length of the amortization period should be established on the basis of the estimated number of years that will benefit from the incurrence of the programming or reprogramming costs. As a general rule the period of amortization of those programs for which there appears to be a continuing need should not exceed the anticipated useful life of the computer. A shorter amortization period should be used in those cases where the contractor can demonstrate by historical data or otherwise that the useful life of the program is shorter than that of the computer. At the larger computer centers, where numerous programs may be involved, an averaging of the expected lives of various programs may be acceptable when such procedure results in a reasonable amortization of the related programming costs over the years benefited.

11-2.4 Amortization Method

The method used to amortize the costs over the estimated useful life of the program should be based on the contractor's normal practice applicable to other items of software. Where this is not possible, any reasonable method of amortizing such costs over the estimated useful life of the program should be considered acceptable particularly if the method is the same as that used for depreciating the equipment.

11-2.5 Justification for Immediate Charging to Current Operations

In some circumstances, the contractor may represent that the desired objective of capitalization/amortization as outlined above is substantially and consistently achieved by charging to current operations all programming and reprogramming costs when and as they are incurred. Due consideration should be given to such representation, provided the contractor submits sufficient data in support of the representation.

11-3 Accounting for the Costs of Computer Software for Internal Use (SOP 98-1)

11-3.1 Applicability of SOP 98-1

On 4 March 1998, the Accounting Standards Executive Committee (AcSEC) of the AICPA issued Statement of Position (SOP) 98-1, Accounting for the Costs of Computer Software Developed or Obtained for Internal Use. In the absence of coverage in FAR, CAS, or other Government regulations, Generally Accepted Accounting Principles will be used for contract costing purposes. All contractors, except state and local governments, will follow the provisions of SOP 98-1, effective for fiscal years beginning after December 15, 1998.

11-3.2 Major Requirements of SOP 98-1

a. Characteristics of Internal-Use Computer Software. SOP 98-1 defines internal-use software as software having both of the following characteristics:

- the software is acquired, internally-developed, or modified solely to meet the entity's internal needs; and
- during the software's development or modification, no substantive plan exists or is being developed to market the software externally.

b. Capitalize Versus Expense. SOP 98-1 stipulates that capitalization of costs should begin after both of the following have occurred: (1) management, with the relevant authority, authorizes (implicitly or explicitly) and commits to funding a computer software project and believes that it is probable that the project will be completed and the software will be used to perform the function intended; and (2) conceptual formulation, evaluation and selection of possible software project alternatives (referred to as the "preliminary project stage") have been completed. After completion of the preliminary project stage, the project proceeds to the "application development stage." Costs related to this stage are capitalized. The application development stage generally includes:

- Designing the chosen path, including software configuration and software interfaces;
- Coding;
- Installation to hardware; and
- Testing, including parallel processing phase.

The costs of data conversion from old to new systems, such as purging or cleansing of existing data, reconciliation or balancing of the old data and the data in the new system, creation of new/additional data, and conversion of old data to the new system, should be expensed. Costs to develop or obtain software that allows for access to or conversion of old data by new systems should be capitalized.

Capitalization should cease when a computer software project is substantially complete and ready for its intended use. Computer software is ready for its intended use after all substantial testing is completed. Costs incurred during the post-implementation/operation stage, such as maintenance and training costs, should be expensed as incurred. The SOP states that even if training cost is incurred during the application development stage, it should be expensed as incurred.

Costs of significant upgrades and enhancements to internal-use computer software should be capitalized if it is probable that those expenditures will result in significant additional functionality. Additional functionality is defined as changes to the software so that it may perform a task it is not currently able to perform.

c. Capitalizable Costs. The following costs incurred during the application development stage should be capitalized:

- External direct costs of materials and services consumed in developing or obtaining internal-use computer software, such as costs incurred to obtain computer software from third parties;
- Payroll and payroll-related costs for employees who are directly associated with and who devote time to the internal-use computer software project, to the extent of the time spent directly on the project.
- Interest costs incurred while developing internal-use computer software (See 11-3.3).

d. Component Accounting. SOP 98-1 applies to the individual components or modules of the computer system. For each component or module of a software project, amortization should begin when the component or module is ready for its intended use, even though the entire software system will not be completed until a later accounting period.

e. Amortization Method. SOP 98-1 provides that capitalized costs should be amortized over the useful life of the software on a straight-line basis unless another systemic and rational basis is more representative of the software's use. For example, accelerated methods of amortization may be appropriate when the utilization of the software is significantly greater in the earlier years of the useful life than the later years.

11-4 Accounting for Costs Related to Enterprise Resource Planning (ERP) Systems

11-4.1 Introduction

Many contractors are investing significant resources in implementing Enterprise Resource Planning (ERP) systems to reengineer their business processes and to replace legacy systems that no longer meet their needs. A typical ERP project involves reengineering business processes and selecting and implementing commercially available software packages from the vendors such as SAP, Oracle, Deltek, etc. This section provides guidance on accounting treatment of cost related to ERP systems. (See 5-406.7 for guidance related to audit of ERP systems internal controls.)

11-4.2 Applicability of ASC 720-45 (formerly EITF Issue No. 97-13) and ASC 350-40 (formerly SOP 98-1)

Financial Accounting Standards Board (FASB) Emerging Issue Task Force (EITF) Issue No. 97-13, Accounting for Costs Incurred in Connection with a Consulting Contract or an Internal Project that Combines Business Process Reengineering and Information Technology Transformation, dated November 20, 1997, addressed the issue of business process reengineering activities. EITF Issue No. 97-13 set forth the typical activities of a business process reengineering project that is part of a broader software implementation project, such as an ERP project. It also incorporated AICPA

Statement of Position (SOP) 98-1, Accounting for Costs of Computer Software Developed or Obtained for Internal Use, which was finalized on March 4, 1998, on internal-use software as guidance on accounting for the software elements of the information technologies transformation projects. The FASB incorporated all prior Generally Accepted Accounting Principles (GAAP) into a single authoritative codification called the FASB Accounting Standards Codification (ASC) for financial statements issued after September 15, 2009. EITF Issue No. 97-13 was incorporated into ASC 720-45 (Other Expenses – Business and Technology Reengineering) and SOP 98-1 was incorporated into ASC 350-40 (Intangibles – Goodwill and Other, Internal – Use Software). In the absence of specific coverage in FAR, CAS, or other Government regulations, Generally Accepted Accounting Principles, including ASC 720-45 (formerly EITF Issue No. 97-13) and ASC 350-40 (formerly SOP 98-1), are the principles contractors must use in accounting for costs related to implementing ERP systems for contract costing purposes.

11-4.3 Business Process Reengineering – ASC 720-45 (formerly EITF Issue No. 97-13)

a. ASC 720-45 (formerly EITF Issue No. 97-13) provides that the cost of business process reengineering activities, whether performed internally or by third parties, is to be expensed as incurred. This also applies when the business reengineering activities are part of a project to acquire, develop, or implement internal-use software. As provided in ASC 720-45-25-2, the costs associated with the following business process reengineering activities shall be expensed as incurred:

(1) Preparation of request for proposal.

(2) Current state assessment: The process of documenting the entity's current business process, except as it relates to current software structure. This activity is sometimes called mapping, developing an "as-is" baseline, flowcharting, or determining current business process structure.

(3) Process reengineering: The effort to reengineer the entity's business process to increase efficiency and effectiveness. This activity is sometimes called analysis, determining "best-in-class," profit/performance improvement development, or developing "should-be" processes.

(4) Restructuring the work force: The effort to determine what employee makeup is necessary to operate the reengineered business processes.

b. Because ERP projects combine internal-use software, governed by ASC 350-40 (formerly SOP 98-1), and business reengineering activities, governed by ASC 720-45 (formerly EITF 97-13), it is important to properly classify such activities. Some of the reengineering activities could be occurring concurrently with software implementation. In order to expense costs as reengineering activities, the focus of the activities should be on process rather than software systems. This is true even if

contractor employees, outside consultants, or software vendors involved in these activities may have information technology and software application expertise.

c. When contractors use an outside consultant or a software vendor to complete an ERP project, the total price of the contract may include multiple elements, such as business process reengineering, software costs, training, maintenance support, etc. ASC 720-45 (formerly EITF Issue No. 97-13) provides that the cost should be allocated to each element based on the relative fair values of those separate activities, not necessarily the separate prices stated within the contract for each element. This is important because some of these costs are required to be capitalized as discussed in 7-105.4 below. The information such as vendor price lists, price charged or quoted by similar vendors, or vendor pricing sheets (rates per hour times budgeted hours) can be used to determine the separate activity market prices. Auditors should ensure that the estimate of fair value assigned to each activity is reasonable and that contractors have adequate procedures to allocate the consulting costs between business reengineering activities and internal-use software development activities (i.e., preliminary, application development, and post-implementation).

11-4.4 Computer Software Developed or Obtained for Internal Use – ASC 350-40 (formerly SOP 98-1)

The software element of ERP projects should be accounted for in accordance with ASC 350-40 (formerly SOP 98-1) which requires companies to capitalize and amortize many of the costs associated with developing or obtaining software for internal use. A typical ERP project encompasses a wide range of software related activities, such as software acquisition, configuration, modification, data conversion, maintenance, etc. Accounting treatment of those activities should be determined based on the criteria specified in ASC 350-40 (formerly SOP 98-1) as discussed in 7-104.

If a contractor has a software license and software maintenance contract from an ERP vendor, the software license costs are capitalized, while the software maintenance portion of the contract is expensed.

ERP systems generally involve several modules or components. ASC 350-40 (formerly SOP 98-1) applies to the individual modules or components of the computer system. For each component or module of a software project, amortization should begin when the component or module is ready for its intended use, regardless of whether the software will be placed in service in planned stages that may extend beyond the reporting period. Auditors should ensure that contractors separately account for costs by module or component to comply with this requirement. Computer software is ready for its intended use after all substantial testing is complete. If the functionality of a module is entirely dependent on the completion of other modules, amortization of that module should begin when both that module and the other modules upon which it is functionally dependent are ready for their intended use.

11-5 Accounting for Costs of Computer Software to be Sold, Leased or Otherwise Marketed – ASC 985-20 (formerly FASB No. 86)

ASC 985-20, Software, Costs of Software to Be Sold, Leased, or Marketed (formerly FASB Statement No. 86 "Accounting for the Costs of Computer Software to be Sold, Leased or Otherwise Marketed"), specifies the financial accounting treatment for the costs of computer software sold, leased, or otherwise marketed either as a separate product or as a part of another product or process. FASB Statement No. 86 was incorporated into ASC 985-20 for financial statements issued after September 15, 2009. ASC 985-20 (formerly FASB No. 86) identifies the point in time that research and development costs incurred in the process of creating a software product to be sold, leased, or otherwise marketed become production costs which should be capitalized and amortized over future sales.

ASC 985-20 (formerly FASB 86) provides that costs incurred internally in creating a computer software product are to be charged to expense when they are incurred as research and development until "technological feasibility" has been established for the product. Technological feasibility is established when either (1) the detailed program design has been completed or (2) a working model has been developed. After technological feasibility has been established, all software production costs are to be capitalized and reported on the financial statements at the lower of unamortized cost or net realizable value and are to be amortized based on current and future revenue. Capitalization of software costs shall stop when the product is available for general release to customers.