

**IT Software Capitalization** – Purpose: To provide guidance for the accounting of costs incurred in a software purchase and/ or development and implementation of software.

1. Phases of Software Development for Capitalizable Software
2. University Owned Software Purchases
3. Software Development
  - a. Non-Cloud Based Software
  - b. Cloud-Based Software
4. Examples of Capitalizable and Non-Capitalizable Costs
5. Software depreciation
6. Software Impairment

### **1. Phases of Computer Software Development for Capitalizable Software**

Operating software, either purchased outright or developed internally, with a cost in excess of \$100,000 as determined by these guidelines shall be capitalized when placed in service. When new software is purchased and developed for specific use by the University, the following phases generally occur:

#### **A. Preliminary Project Stage**

Also known as the planning stage, this portion of the project is used for researching software and making decisions to move forward with the purchase of software. Costs are expensed as they are incurred in this stage. Examples of expenses incurred in the preliminary phase are as follows:

- Determining a need for new software or upgrades to existing software
- Discovering software that will meet business needs
- Vendor/ product review and selection (vendor demonstrations to ensure software will meet the needs of the organization)
- Creating a team - obtaining consultants, project managers, software and functional area experts, and backfills if needed

#### **B. Application Development Stage**

Once software and a team are selected and funds are committed to the project (contracts are signed), the preliminary stage is complete and the application development stage begins. In general, costs in this phase **are eligible to be capitalized**. This phase includes the following work:

- Designing/ configuring software
- Coding /customizations
- Installation of software and hardware – see additional information in ‘Software Development’ section below on Cloud Based software vs. Non-Cloud based software capitalization requirements
- Quality assurance testing

#### **C. Post-Implementation**

Also known as the operations stage, this is after the new software has been substantially installed throughout the organization and is being used by the majority of users. There may

be a stabilization period where changes to the system occur, this is still considered to be post-implementation. In general, this phase means all testing has been substantially completed and the new software is ready for its intended use. Costs in this phase are expensed as incurred and include the following types of activities:

- Data conversion costs
- Training
- Application maintenance (over the maintenance period)
- General & administrative costs
- Overhead allocation

Note: Costs in each stage may occur throughout the project. For example, training (post implementation phase) may be occurring at the same time as quality assurance testing (application development phase). In situations such as these, apply the policy above to the nature of the costs instead of the timing. Therefore, training costs would be expensed while quality assurance testing would be capitalized.

Additionally, if software can be used out of the box (development costs are not needed), the above phases may not all apply. See software purchases section below for more information on these purchases.

## **2. University Owned Software Purchases**

Per policy 1004.1 [http://finserv.uchicago.edu/support/policies/1000/1004\\_lbeb.shtml](http://finserv.uchicago.edu/support/policies/1000/1004_lbeb.shtml), operating software included in the price of the hardware will be capitalized as long as it meets equipment requirements (ex. >\$5K and > 1 year useful life). When purchased separately, software will be capitalized if the cost exceeds \$100,000 (excluding maintenance, support, and training costs). Software license subscription fees are expensed as they are assumed to be a fee that is paid to use the software, but the software is not actually owned. In order to be capitalized, the University must have the contractual right to take possession of the software and be able to run the software on our hardware. For software invoices that include both capital and noncapital costs, the unit should make an attempt to split the respective costs when coding the invoice and maintain adequate documentation on the split.

## **3. Software Development**

As previously stated, hardware and software that are purchased are capitalizable per the above section. However, often there are additional expenditures incurred to develop/ implement the software. Per policy 1004.1 (see link above), *internally developed software* with material and labor costs in excess of \$100,000 will also be capitalized (see point 4 below). For university owned equipment, incremental costs that add to the future value of the asset are considered capital. Expenditures that do not increase the value of the asset are expensed. See specific examples of capital and non-capital costs in section 4.

### **A. Non-Cloud Based Software**

In non-cloud based systems where the University owns the system, software costs considered for capitalization. Any updates made that add functionality to the software or

increase the software's useful life can also be considered for capitalization, and the project phases and capital categories listed in this document would apply.

### ***B. Cloud Based Software***

In cloud based software development (sometimes referred to as a hosting arrangement), the software could be capitalized if the following items are met:

1. The University has a contractual right to take possession of the software at any time during the hosting period without significant penalty (either significant cost or significant loss of value to the software/ data)
2. It is a feasible option that the University could run the software on its own hardware or find another unrelated vendor to host the software.

If both of these criteria are met, are acted on (or probable expectations to act on them), and support can be provided, you would be able to follow the non-cloud based software development section above.

Absent meeting these requirements, the University would pay a subscription fee to access the software via the cloud. The actual software subscription fee should be expensed when incurred. However, the labor and expenses to implement the new system are capitalizable during the implementation phase only, provided they meet the requirements discussed above.

Note: If systems are developed at the same time, they may be considered one asset. For example, if a new financial services system was in development and it included an accounts payable sub ledger and an accounts receivable sub ledger, they could be considered 3 separate components of the same system or if they were developed at the same time they could all be included as one Financial Services System asset.

#### 4. Examples of Capitalizable and Non-Capitalizable Costs

The following are examples of costs that are considered capital.

<b>Capitalizable Costs - Application Development Phase Only</b>	
<b>Category</b>	<b>Examples/ Additional Information</b>
Design Fees	<ul style="list-style-type: none"> <li>• Software Configuration</li> <li>• Software Interface</li> <li>• Coding</li> </ul>
Internal Salaries & Benefits	<ul style="list-style-type: none"> <li>• Salaries, benefits, bonuses, and food may be capitalized for the following two groups of people:               <ol style="list-style-type: none"> <li>1. New employees spending 100% of their time on the implementation project; and</li> <li>2. Backfills who replace employees for the lifespan of the project</li> </ol> </li> </ul> <p>Meals must be extraordinary in nature in order to be capitalizable. DOES NOT include morale related costs such as team meals out and alcoholic beverages.</p>
External Consultant Fees	<ul style="list-style-type: none"> <li>• Consulting fees incurred during application development are capital.</li> </ul>
Travel	<ul style="list-style-type: none"> <li>• As long as travel is 100% project related (and not training related), it can be capitalized</li> </ul>
Hardware/ Installation of Hardware	<ul style="list-style-type: none"> <li>• Similar to purchased/ out of the box software, hardware required to make the software function as necessary is considered capital.</li> </ul>
Conversion Costs	<ul style="list-style-type: none"> <li>• Software developed or purchased to allow old data to be accessible by new software</li> <li>• All other conversion costs are expensed. See below</li> </ul>
Testing of new software	<ul style="list-style-type: none"> <li>• Ensuring software is working as intended prior to implementation</li> </ul>

<b>Capitalizable Costs - Outside of Application Development</b>	
<b>Category</b>	<b>Examples/ Additional Information</b>
Upgrades/ enhancements	<ul style="list-style-type: none"> <li>• Any upgrades or enhancements that increase functionality or extend the useful life of the hardware and software.</li> </ul>
Hardware/ Installation of Hardware	<ul style="list-style-type: none"> <li>• If this meets independent equipment requirements, it will be capitalized as equipment.</li> </ul>

The following are examples of costs that are considered non capital/ expenses:

<b>Non-Capitalizable Costs (Expenses)</b>	
<b>Category</b>	<b>Examples/ Additional Information</b>
Determining Requirements	<ul style="list-style-type: none"> <li>• Time/ resources used in determining a new software implementation is needed vs. software upgrade</li> </ul>
Vendor/ Product Evaluation	<ul style="list-style-type: none"> <li>• Time and expenses required to determine what software will be implemented and what vendor/ consultants will be employed</li> </ul>
Data Migration	<ul style="list-style-type: none"> <li>• Purging/ Cleansing old data</li> <li>• Reconciling data</li> </ul>
Training	<ul style="list-style-type: none"> <li>• Trainer salaries and benefits</li> <li>• Training materials</li> <li>• Travel related to training</li> </ul>
Maintenance	<ul style="list-style-type: none"> <li>• General maintenance to keep system running after go live</li> <li>• Annual software maintenance fees/ licenses</li> <li>• Upgrades that do not increase functionality/ useful life</li> <li>• Minor upgrades to keep software up to date</li> </ul>
Administrative Costs	<ul style="list-style-type: none"> <li>• Overhead fees</li> <li>• Space</li> <li>• Utilities</li> <li>• Office supplies</li> </ul>

## **5. Software Depreciation**

Software (both purchased and internally developed) is given a useful life of 5 years unless a specific useful life is identified and communicated to Capital Asset Accounting ([capitalassets@uchicago.edu](mailto:capitalassets@uchicago.edu)). Depreciation would begin in the year in which the system has been successfully tested following the University's equipment half-life depreciation guidelines. Refer to policy 1004.2 for additional information on the University's depreciation policy ([http://finserv.uchicago.edu/support/policies/1000/1004\\_lbeb.shtml](http://finserv.uchicago.edu/support/policies/1000/1004_lbeb.shtml)).

## **6. Software Impairment**

Software costs may need to be written off if it is probable that the software that is being developed will no longer be complete/ placed in service or if the completed software has become obsolete.

Source for information is interpreted from Accounting Standards Codification (ASC) No. 350-40 and Statement of Federal Financial Accounting Standards (SFFAS) 10.