WHITE PAPER

SOFTWARE ASSET MANAGEMENT: PRACTICES FOR SUCCESS

SAM tools are of little value without processes and people to support them.

Executive Summary

All organizations require software systems to conduct daily operations. These strategic enterprise assets are often acquired or created in isolation of each other as an organization grows, particularly for those that provide services rather than physical products. Over time, these systems become complex, overlapping and highly dependent on one another, so more effort must be expended to discover the full effects of any system change.

The ability of an organization to update, replace or improve the overall capabilities of the enterprise can become severely impaired. Moreover, support and administrative costs for these critical assets inevitably grow to be a major burden. Add to all this the need to control licensing, follow the organization's policies and ensure security for the internal network, and it becomes clear that some manner of oversight is required for the enterprise to be successful.

TWEET THIS!

Table of Contents

The practice of software asset management (SAM) addresses these cost-generating issues. As with any valuable enterprise asset, software represents a large, ongoing investment of time, capital and personnel. It is, therefore, necessary to develop and maintain a controlled inventory of these assets using a well-managed set of tools and best practices.

The goal of SAM is to develop this inventory, determine the value of each asset to the organization, manage licensing for purchased software products and provide a mechanism for the timely, deliberate update of software systems as the organization grows. The success or failure of these efforts is measured by the ability to match the right tools and practices to a particular organization's SAM needs.

The introduction of a SAM program affects both strategic direction and tactical operations. The information contained in the SAM inventory contributes to determinations of costs and can affect strategic plans when an organization is considering mergers or acquisitions, opening new markets, rolling out new products or services, introducing new policies and procedures or making other decisions.

Likewise, a SAM team contributes directly to the operations of an organization by tracking the use of software licenses and the acquisition of new software systems, as well as defining the effect that system upgrades or replacements may have on operational costs.

SAM Superlatives

An enterprisewide software asset management program provides three primary benefits: governance, management and enterprise agility.

Software system governance focuses primarily on ensuring that only approved applications are introduced into the local computing environment and that software licenses are fully and appropriately utilized. To this end, SAM will provide automated scanning tools that enforce the organization's policy regarding unapproved or pirated software.

These tools provide the SAM group with information on every networked computer within the enterprise, including the application name, location, date of installation and other critical information that can be used to identify and remove unwanted software. The advent of mobile devices connecting to wireless networks complicates this picture, with malicious applications on mobile devices increasingly threatening network security.

Finally, with the advent of cloud-based computing (either platform as a service or software as a service), the responsibilities of the SAM team have expanded to include security for software that is not even physically present in the enterprise.

The second primary benefit of SAM is facilitating the overall management of the software assets themselves. Software represents a significant financial investment affecting capital expenditures for research, acquisition and deployment. Also, SAM helps control operational expenditures for ongoing license costs, upgrade costs, manufacturer support and the effect on operations during outages.

For license management, the SAM program is used to measure license utilization (or underutilization) and to respond by acquiring new licenses, negotiating for better license rates, transferring unused licenses to other clients or groups, or canceling unused licenses for cost savings. SAM is also responsible for defining the license request process and for researching appropriate tools to meet the needs of individuals and groups within the organization.

In addition to simply noting the number, type and cost of each software system, the SAM program is also responsible for determining the proper timing of upgrades, patches and other maintenance activities. Moreover, before any new systems are built or purchased, the SAM team should be consulted to see if existing systems can provide the desired function (perhaps with customization or a modification request to the software maker). In this way, duplication of software functions is controlled, especially in larger enterprises where visibility across the software footprint is otherwise difficult to achieve.

SAM's operational function also extends to system and network security. Every year, billions of dollars are lost to network intrusions, data theft or other forms of automated attack. Often, these attacks are enabled by the organization's workforce, through actions such as downloading a seemingly innocent email attachment that leads to significant damage.

The focus of the SAM's security contribution is on detecting and removing unapproved software, as well as illegal (or unlicensed) software utilization. Even when an organization is careful to limit users' ability to access and modify computing assets, it is still possible for individuals to install a "favorite" application or utility. This opens the organization to legal risk for piracy and security risk for attack. In both cases, SAM automated tools are used to defend against these risks.

Finally, the SAM program provides the organization with improved agility and the ability to respond more rapidly to changes in the marketplace. Specifically, by tracking all of the applications in use by the organization, and providing advice to senior management on applications that will or will not work well within the current environment, the SAM team helps make better strategic and tactical decisions regarding software deployment.

The Many Flavors of Software Licensing

The many forms of software licensing require different techniques for managing the acquisition, assignment, reclamation and retirement of each license. The following table includes some of the most common forms of software asset licensing:

License Type	Description	Management Style	Example
Identity-locked	A form of license is sold on an individual basis and assigned to a specific user. Once used, a license is expended and is no longer available or must be released by the provider to be reassigned to another named user.	Identify all individuals who have consumed the license for a particular software system. If a license can be reassigned or reclaimed, note the proper method for updating the software provider.	Microsoft Windows
Node-locked	A type of license that's registered with a particular computer system or processing node. Although multiple users may have access to the system, only one licensed version may be running at any time.	Like identity-locked license keys, a node-locked key is typically a single-use, single-machine key. But some software makers allow reassignment of a key as long as only one instance of the software is using the key at a time. Treat these keys like assignable assets in the SAM inventory.	Database software systems, desktop applications
Processor-locked	These licenses are assigned based on the number and type of computing processors in the host machine.	These licenses are typically limited to centralized computing resources accessed by a large user population. They should be tracked in the SAM inventory for the number and location of software installations.	Mainframe applications, database servers, application servers
Floating	Concurrent use of licenses that allows a limited number of individuals to use a key from a defined pool.	A central network server often administers floating licenses. In the SAM inventory, note the type, number and registered software version, as well as the server that hosts the license manager.	Shared desktop applications, network-accessed software systems
Site license	Unlimited usage based on the site of the software installation.	These licenses are usually handled on an annual renewal basis and do not typically need to be tracked by individual utilization.	Desktop applications, utility software
Open-source	Unlimited usage within the limits set by the open-source agreement.	There are many different agreement types for open-source software. The SAM team should consult with the legal department to determine what, if any, limitations exist and must be managed.	Open-source software programs
Limited-use	A rarely used license form in which the software provider limits the purpose to which the software is applied.	This form of license requires the SAM team to validate the installation and use of the specified software systems.	Hardware-embedded software
Unlimited-use	The software provider grants unlimited use of the software but retains copyright controls over modification.	The SAM team tracks the installed software base and any support agreements, but license tracking is not required.	Public-domain software
Timed	The license restricts the software system to a limited time, after which the software is unusable; often associated with individual, node-locked or floating licenses.	These licenses are tracked depending on the specific type of license (i.e. floating), the expiration date and the software system that is licensed.	Varies
Physically restricted	These types of licenses are rare but use a physical device (such as physically attached hardware) to enforce licensing restrictions.	The license is represented by a physical asset (i.e. a dongle) that is tracked in a similar manner to other computing resources.	Dongle-enabled software systems

The SAM team is directly responsible for recommending the replacement or retirement of software systems, a responsibility directly tied to the enterprise's ability to introduce new system functionality, replace aging or unsupported systems and reduce licensing costs by changing vendors. Over time, these improvements can become apparent through improved customer service and retention (as systems become more stable and resistant to unexpected outages), timely reporting on system operation and maintenance costs, and a reduction in redundant applications.

The SAM Dream Team

There are a number of wide-ranging considerations for organizations to think about when establishing a SAM practice. The first is the cost of oversight and tool acquisition. Setting up a trained team of SAM professionals dedicated to the discovery, capture and management of software assets requires the investment of time, money and organizational will. However, the payoff in governance, management and enterprise agility is well worth the cost.

A SAM team will touch many different areas: network and application security, efficient and reliable platforms, control over acquisition and maintenance costs, improved time to market for new services, and the ability to take advantage of cloud-based applications.

Many SAM teams fall under the guidance and control of the enterprise architecture group. This practice has several positive aspects, given the overarching nature of the EA team's responsibilities.

Enterprise architects can utilize the software asset inventory to develop software dependency maps. These maps can show relationships between various software systems across the enterprise, as well as the impact of changes to those target systems. Given the tendency over time for applications to interact (especially at the data tier), these maps become critical to SAM's role in replacing or retiring software systems.

Software Ecology

Interacting software systems create their own kind of ecosystem. They rely on each other and are all necessary to the overall health and stability of the organization. As with any ecosystem, they grow organically over time as additional needs are identified that cannot be met by the current software. Unless controlled in some manner, these ecosystems become redundant, expensive to modify and are ultimately very fragile.

Systems in an ecosystem rely on other systems to provide critical operations or data. For example, a particular

Mergers & Acquisitions and SAM

One of the most difficult tasks an organization can face is to merge the operations of a newly acquired unit with its current workings. The two organizations may share a related goal, but each has grown via a different path. This is clear to see when two separate software ecosystems collide.

Software assets are often overlooked until very late in the acquisitions process. This can make it difficult to assign licenses to personnel, who then become unable to perform their newly assigned tasks. Software integration challenges also make it harder to support existing systems long enough for migration to be performed, or to deal with the expiration of key support contracts.

Unfortunately, SAM teams are often the last group consulted during a merger. The existing licenses, software systems and databases from the two organizations are therefore less likely to be properly considered for operational costs and the cost of system migrations. This may lead to a situation where a merger is completed on paper but is far from being realized in day-to-day operations.

Duplication of key systems doubles the cost of support until the data and superfluous software are properly assimilated or replaced with the prevailing software systems, which may not have been properly sized to handle the new user base or processing loads.

Moreover, documentation on license utilization, purchase and pricing history, vendor contact points, support agreements and even the location of backup systems is often scattered, incomplete or simply out of date.

It is typical of organizations that do not have a robust SAM practice to have little knowledge of the current versions of the installed software base, let alone the assignment of licenses or even the location of common license servers.

To address this concern, organizations should conduct a pre-merger software asset assessment. This audit will properly determine the software systems in use in the acquired organization and provide information on license utilization. Records should be examined for purchasing history and current software release levels, particularly with regard to manufacturer-supported systems.

The SAM team should prepare a final assessment, which should include identification of overlapping software functionality, an estimate of current licenses (utilized and available by type of license), support contracts (with termination or renewal dates) and current purchase agreements and present all of this information to the executive team well in advance of the merger finalization.

operation (such as providing for the delivery of an ordered product or service) may encompass multiple software platforms, all of which participate in the execution of that operation. In addition, multiple software systems are used to support internal workflows, such as payroll, accounting or reporting.

These internal programs may participate in a variety of additional ecosystems, which are no less critical than those that are externally facing. Taken together, both internal and externally facing applications require oversight and control to ensure proper and efficient interaction.

Clearly, the health of these software ecosystems directly affects the health of the organization. If they are allowed to degenerate into a tangled mess, with no governance over licensing, proliferation, interdependence or installation of unapproved software — and no way to measure each system's value — then the entire enterprise will suffer.

The situation will likely yield increased operational costs, limited flexibility to react to changing market conditions and even potential legal trouble for inappropriate license management. The organization requires consistent oversight of software ecosystems to avoid these issues.

SAM is an ongoing and evolving process. It requires commitment from all levels of the organization, from the executive management team to each end user. Everyone participates in the process: by ensuring that they do not introduce unwelcome software onto a local computer system, by taking inventory of purchased products or by defining the purpose and goals of each software ecosystem in the organization.

Although SAM practices are supported by a set of automated tools, no tool can substitute for a clear understanding of the drivers and constraints placed upon the SAM team. As such, there are three primary aspects to SAM — review, reconciliation, and reengineering and retooling. We will explore each of these drivers and the effect they have on the selection of tools and best practices for the organization.

Step 1: Review

The first step in the development of a SAM practice is to establish a full inventory of each software system that is deployed. Although in theory this is possible by manually collecting the information from end-user computers and servers, in practice it is performed by an automated survey of deployed software systems. Many manufacturers have created products to perform this task.

Establishing a SAM practice is independent of automated tools, but can be made much easier, depending on tool selection. An initial scan is performed using an automatic discovery tool, such as the IBM Tivoli Configuration Manager or the Microsoft System Center 2012 Configuration Manager, to provide a baseline of the current applications on the network. In many cases, it will be necessary to run multiple scans to ensure that all mobile computers, or other systems that are not always online, will be captured in the initial baseline. It is good practice at this point to establish a recurring process that will periodically scan the entire network and update the software asset inventory on a regular basis.

The finance group should be consulted to determine which software has been purchased and which requires the assignment of a license. Note that software providers offer a wide array of licensing schemes (see *The Many Flavors of Licensing* sidebar, page 3), which will determine the governance processes that will be instituted to control licensing costs and ensure that an adequate number of licenses are available for authorized users.

Once the initial baseline of licensed software is determined, the number of licenses currently in use (depending on the provider's licensing scheme) and the number of free licenses can be determined for each software system. It is common to find that licenses have been assigned to individuals who are no longer using the software or who have left the organization.

How To Categorize Software Systems

The categorization of software by function into a SAM inventory is quite a difficult prospect when starting from scratch. But completing such a project is valuable because it can give an organization the ability to rapidly determine the effect of a system upgrade or the replacement of a particular platform.

To assist in implementing SAM, various generic taxonomies can serve as a starting point. One such structure for telecommunications is the TM Forum's *Application Framework*.

For the purposes of SAM, one approach to organizing applications is to represent the business functions as an ecosystem (including the enterprise), with the business process areas cutting across those ecosystems. Therefore, applications that fall into the intersection of two category dimensions would be named in the SAM inventory as "ECOSYSTEM::PROCESS NAME::APPLICATION." Here are a couple of examples: BILLING::SERVICE MANAGEMENT::APPLICATION NAME or ENTERPRISE::APPLICATION NAME.

For applications that provide multiple functions across ecosystems or process boundaries, the primary function of the software determines the appropriate category structure to which it belongs. For more fine-grained organizing, the *Application Framework* includes subcategories for each business process area. In this case, the lower category would optionally be appended to the application name: ENTERPRISE::ADMINISTRATIVE SERVICES::APPLICATION NAME.

The baseline inventory is then stored in a management tool. Various software makers provide inventory databases to support configuration scanning components. These inventory systems can vary greatly in both features and cost. An organization should carefully consider its needs and the expected growth of its software assets. (A small organization with less than a dozen controlled software programs will not need as sophisticated a management and oversight tool as a Fortune 500 company.)

For each discovered software system, the next task is to determine what features the software offers to the organization and to assign an owner for the system. This is a key step in the review process. The system owner will be the primary contact point for scheduling system updates, fixes and replacements. All of this information is collected as part of the baseline inventory in preparation of the reconciliation phase.

Step 2: Reconcile

Once an inventory has been collected, the next step is to organize the system information to allow for proper analysis. For this phase, each system is assessed for its participation in a given operational ecosystem, which is noted in the inventory. Although there are numerous ways to organize and categorize software systems, many industries have already determined recurring ecosystems for common functions.

For example, many organizations require some form of automated support for human resources, including services for employee tracking, benefits and compensation. Software systems acquired to support these functions would be assigned to an "Enterprise Support" ecosystem.

More complete examples can be found in the *Application Framework* published by the TM Forum, a global industry group, or the IT Infrastructure Library's *Guide to Software Asset Management* (see *How to Categorize Software Systems* sidebar, page 5).

In general, the procedure that most organizations follow to determine the proper category for software systems remains the same:

1. Define the role of each system in the organization's operations.
2. Assign the software to a management group (ecosystem) based on core system functionality.
3. Determine strategic value for each system (for example, critical, essential, important, minor or unnecessary).
4. Settle on an ecosystem manager (from the SAM team) responsible for the oversight, research and acquisition of a specific ecosystem application suite.

Based on this assessment, software assets can be properly managed by individuals most familiar with the function of each application, including the licensing and acquisition needs of each group. Working with application owners (assigned during the review stage), ecosystem managers can establish common practices for software acquisition (including the issuing of requests for proposals or requests for information to contractors), enforce licensing policy and ensure that upgrades are performed on a regular basis.

For systems that require license management, this structure provides support for a controlled assignment and reclamation of licenses. Each licensed application should be noted for the following to avoid overuse or underutilization of these costly resources:

1. Determine the number of licensed applications, and identify information such as software maker, count, cost per license and current license holders.
2. Categorize all license usage: assigned in-use, assigned unused, unassigned or expired.
3. Quantify licensing costs and needs.
4. Assign support level costs and utilization.
5. Identify and remove unapproved software.
6. Determine the effect of removal of software systems.
7. Make note of approved applications that will meet the organization's needs.

Once software assets are discovered and organized, the SAM team can begin the process of replacing aging or obsolete software systems in a controlled and deliberate manner by reengineering.

Step 3: Reengineer & Retool

Software assets have a defined lifespan: They are researched, acquired, utilized, updated and retired. It's the SAM team's responsibility to manage these stages for software assets from the moment a request is made, through purchasing and deployment, maintenance, and finally removal from service. Each step of this process is supported by the SAM inventory previously created by the team, to reduce duplication and to identify applications that are no longer meeting the organization's needs.

The SAM team is tasked with ensuring that all applications are properly upgraded within the software makers' specified support timeframes. Many application providers release updates, fixes, patches and other forms of software maintenance to provide users with an improved system experience. The SAM team, as part of its research function, provides system owners information about these update schedules and the expected improvements to their systems.

If a system owner forgoes regular maintenance, perhaps because of stability issues or outage concerns, then this must be tracked in the SAM inventory for the applications.

Eventually a decision must be made to upgrade the software to a supported level or the organization runs the risk that the maker will refuse to honor a support agreement.

Finally, the SAM team should work with each ecosystem manager and various application owners to determine a retirement or replacement plan for obsolete software systems. These retirements often affect other systems through some form of dependency (either by data or system function), and must be carefully managed through a well-designed enterprise integration strategy.

The Right Tools for the Job

Each SAM solution has advantages and drawbacks as well as widely different costs and support structures. The two proceeding charts on this and the next page provide an initial assessment of basic capabilities offered by each product type and introduce the major and minor players in the SAM space. It is a primary task of the SAM team to determine the best fit of a particular tool to the organization's needs.

Long-term ROI

The management of software systems has become an essential process for almost all organizations — as important to success and achieving a decent return on investment as accounting is to revenue and expenses. The practice of software asset management was developed specifically to provide system and cost oversight. Thus, an organization that can efficiently manage its software assets will have a significant advantage.

SAM goes well beyond simply accounting for purchased software and extends into license management and governance as well as deployment, configuration control and software lifecycle maintenance. A high level of commitment to ongoing SAM efforts will pay handsome returns as the organization's automation needs grow, providing both flexibility in services and a mechanism for controlling costs over time.

SAM Toolkit Features Explained

Feature	Description
Software license management	This feature allows the automated discovery of licensed and unlicensed software on the organization's network, tracks assignments of licenses to individuals and groups, and allows reclaiming of license resources from individuals who no longer require access.
Software configuration	Software is typically deployed with a particular set of configuration settings to defined servers. This feature tracks the configurations in use for each deployed version of a software product.
Software inventory management	All software assets must be tracked in a uniform, centralized manner to maintain proper controls. This feature facilitates the automated capture of software assets from the network and provides for tracking of each asset by owner, purpose, software maker, location and other relevant information.
Software utilization	This feature provides the SAM team with the ability to track and report on overall software utilization, including the number of concurrent users, peak usage and service outages.
Contract and purchasing	For SAM teams that are responsible for contract and purchase management of software assets, this feature helps the organization manage each software contract and links the SAM inventory to the organization's accounting and purchasing systems.
Compliance and governance	This feature tracks unapproved use of software on individual platforms, such as desktop or notebook systems. Reporting provided by this capability allows the SAM team to identify violations of policy and to ensure compliance among users.
Software lifecycle management	This feature oversees the full lifecycle management of a software program, starting with a request for proposals and continuing through purchase, acquisition, deployment, maintenance, sunsetting and retirement.
Software deployment (local, server, virtual, cloud, mobile)	The SAM team can track the deployment environment of each software system, including specific hard, virtual or cloud-based platforms. This feature is often used in conjunction with software configuration management.
Reporting, measurement and analysis	This feature provides a suite of reports based on information obtained from other aspects of the SAM practice. This information may be presented as generated reports, dashboards or data warehouses.

Comparing SAM Toolkits

SAM Tool	Licensing	Configuration	Inventory	Governance	Contracting	Lifecycle
CDW Software Asset Manager	X	X	X	X		
CDW Software License Manager	X		X	X		
Express Metrix	X		X	X	X	
FrontRange Software Asset Management		X	X			
IBM Tivoli Asset Management	X	X	X	X	X	X
LANDesk Asset Lifecycle Manager	X		X			X
Microsoft Assessment and Planning Toolkit*	X	X	X			
Microsoft Asset Inventory Service			X	X		
Microsoft System Center 2012 Configuration Manager		X				
Novel ZENworks Asset Management	X		X	X	X	
Symantec Altiris Asset Management Suite		X	X			

*For use with Microsoft products only

Microsoft® System Center is a comprehensive management platform that enables you to more easily and efficiently manage IT environments, including server infrastructure and client devices. With System Center, you get a cost-effective and flexible platform for managing traditional data centers, private and public clouds, and client computers and devices.

CDW.com/microsoft

The ideal balance between user flexibility and readiness, LANDesk® Management Suite gives you all the control you need – no matter how big or diverse your environment – to address IT concerns throughout the organization. It enables you to discover devices in your network and store information on its configurations, OS, processor speed, installed memory and more – in a central database.

CDW.com/landesk

Symantec™ Altiris™ Asset Management Suite improves visibility into your IT assets at every point in the lifecycle to reduce costs and fulfill compliance initiatives. The suite helps you eliminate unnecessary software and hardware costs, proactively manage vendor contracts, and align resources to ensure IT investments are optimized and aligned.

CDW.com/symantec

Maintaining network infrastructure is more critical than ever. Enterprise networks are no longer static. Technology shifts are forcing the network to become dynamic and the result is dramatically changing how customers need to perform network management. With IBM® Network Management you can take control of mobile devices, smarter infrastucture, wireless, VoIP, clouds, virtualization, Big Data and video conferencing.

CDW.com/ibm

TWEET THIS!

PEOPLE WHO GET IT™