White Paper

Taking an Asset Management Approach to Reusable Technology Building Blocks What to Buy, Hold and Sell

WP0092 | July 2013

Guy B. Sereff

Guy Sereff is an author, speaker and technology practitioner. His Technology Industry experience includes Application Research and Development, Large-Scale Technology Management, and Global Enterprise Architecture.

As well as a pragmatic blend of Strategy and Tactical execution, Guy also has extensive Architectural Domain experience which covers Business Architecture, Information Architecture, Solution Architecture and Enterprise Architecture.

Nearly every organization of every size has amassed a heterogeneous array of technology gadgets and an equally wide assortment of software packages, utilities and applications over the course of operations. The pace of today's global economy makes it very difficult to stay on top of the vast number of technical components deployed across a firm, let alone with an effective strategy or game plan. Legacy tools and technology elements get woven deeper and deeper into core business processes over time, often in obscure or unclear ways that make them difficult to manage well, if at all; yet this collection of 'widgets' often represents years of what was, at the time, a thoughtful investment in meeting the immediate needs of the organization.

Considering all of the components as assets allows us to apply common asset portfolio management techniques. By following a basic three-point plan of (1) Organize, (2) Categorize, and (3) Optimize, firms can focus their energies and investments on performing assets (i.e. adding or sustaining value) and divest themselves of assets that are under performing (i.e. marginal or declining value). Many firms do a great job of getting through the first step, and a pretty good job at the second. However, it can be difficult to master the last step and reap the full benefits of asset optimization. In this paper we'll discuss how using an asset Buy/Hold/Sell strategy across the asset portfolio can help drive such optimization, and facilitate progress against the firm's Enterprise Reference Architecture roadmaps.

Architects, business analysts, product managers and IT managers will benefit from this approach by linking together best practices around Software Asset Management (SAM), Information Technology Asset Management (ITAM), Architectural Building Blocks (ABB), System Building Blocks (SBB) and capability modeling. Identifying patterns of potential reuse and platform simplification through asset optimization should help map the path forward, accelerating the implementation of the target reference architecture.

ORGANIZE
OPTIMIZE
CATEGORIZE

Step 1: Organize

As the late Dr. Covey taught us, it is best to begin with the end in mind.[i] For our purpose within the scope of this discussion, that end is the implementation of a process that continually optimizes the entity's technology assets across the enterprise, with a keen eye towards knowing which assets to add to, maintain in, and liquidate from the portfolio. Before we can make progress, we have to get a handle on what our technology asset portfolio looks like. This leads us to our first logical step, which is to organize our core assets in a meaningful way. Taking shortcuts or looking at only a cross-section of our technology asset portfolio may provide some marginal lift or directional insight, but organizations are advised to take a deep cut at identifying their total asset portfolio.

What is an Asset?

The term asset can be a bit overloaded these days, so we'll use the following definition to baseline our discussion:

An asset is a discrete, identifiable bounded element that provides some form of value to the entity or organization, whether that value is real or perceived. Obvious technology asset types include software, hardware, storage devices and communications equipment. Less obvious asset types include capabilities, business components and business processes. Sereff, 2013

Generally we find that identifying technical assets makes sense to us; they are intuitive and generally easy to identify, so we often tend to gravitate towards them as a starting point when establishing an asset inventory. But the suggestion here is to consider both the capabilities and the supporting processes as assets as well. We'll discuss the linkage more when we get to Step 2. There are many ways to approach organizing the firm's technology assets, including Software Asset Management, or SAM as defined by ISO/IEC 19770 and Information Technology Asset Management, or ITAM as outlined in the ITIL specifications. Let’s explore these two standards as a means of jump-starting the institution’s asset organization efforts.

SAM - Software Asset Management

Several years ago the concept of Software Asset Management emerged as an initial attempt to gain some level of understanding and control of what software assets were active across a firm. Such knowledge would help system administrators identify which vendor packages were over licensed (bought more than actual use required) and which packages were under licensed (overused whether overtly or unintentionally). Too many licenses may be good for vendors but can be a hidden ‘tax’ on the firm’s operating budget. Too few licenses may save a few hundred Euros here and there in the short run, but incur legal action based upon the pervasiveness of the violations.

ISO/IEC 19770

To that end, the International Organization for Standards (ISO) and the International Electrotechnical Committee (IEC) has published ISO/IEC 19770, which speaks directly to the practice of Software Asset Management.

“ISO/IEC 19770 has been developed to enable an organization to prove that it is performing Software Asset Management (SAM) to a standard sufficient to satisfy corporate governance requirements and ensure effective support for IT service management overall.”[ii]

ISO/IEC 19770 consists of three main parts, each addressing a core aspect of managing software assets. As with most standards, it is not really prescriptive in terms of describing how something should be done. Instead the focus is on what should be done; outlining a best-practice template to follow for sufficient topical coverage and a mechanism for assessing practice maturity.

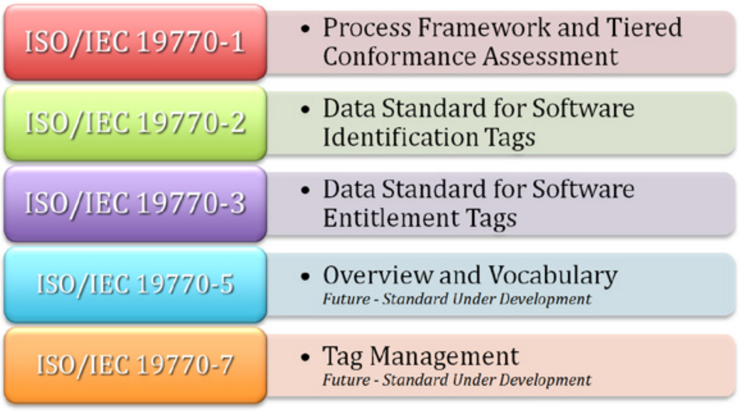
ISO/IEC 19770-1
• Process Framework and Tiered Conformance Assessment
ISO/IEC 19770-2
• Data Standard for Software Identification Tags
ISO/IEC 19770-3
• Data Standard for Software Entitlement Tags
ISO/IEC 19770-5
• Overview and Vocabulary
Future - Standard Under Development
ISO/IEC 19770-7
• Tag Management
Future - Standard Under Development

Figure 1: ISO/IEC 19770 Sections[iii]

The first section of the ISO/IEC 19770 standard deals with ensuring that the organization has established a formal process for governing their software assets and to underscore the firm’s effective software service support model. Section 1 also includes guidelines for assessing conformance to the process in a tiered maturity model. This checklist is an excellent starting point when either creating or assessing the entity’s software asset management approach. The second and third sections relate to establishing data tags or ‘markers’ to identifying software instantiation and end-user entitlements. Sections 5 and 7 (yes, there are no sections 4 and 6) are in progress and will focus on standardizing SAM taxonomies and software asset tag management. Note that ISO/IEC 19770 is generally geared more towards vendor software application packages to protect both the supplier and the consumer. However, these same SAM concepts can (and should) be equally applied to custom packages and proprietary applications that have been developed in-house. Our goal is to manage all of our software assets – not just those of interest to our software vendors and suppliers and the *BSA | The Software Alliance organization*.

As with the application of any standard, there is a maturity assessment model available to help institutions to understand how mature the practice is relative to the standard. The Software Asset Management Standards Working Group has identified four specific levels of maturity. Tier 1 is about establishing confidence and validity in the asset inventory data. Tier 2 represents establishing practical or pragmatic control of the asset inventory. Tier 3 represents the integration of software asset management into the operating procedures. The pinnacle, Tier 4 is achieved when the organization is in full compliance of the ISO/IEC 19770 standard, fully conforming to the guidelines therein.

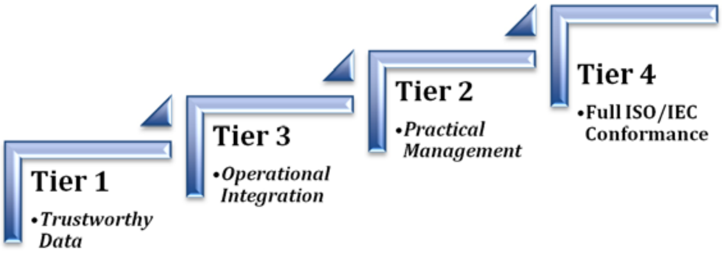

Figure 2: ISO/IEC 19970-1 SAM Maturity[iv]

Information Technology Infrastructure Library

The Information Technology Infrastructure Library (ITIL) defines SAM among its various process governance domains. From the ITIL perspective, SAM is defined as:

> *…all of the infrastructure and processes necessary for the effective management, control and protection of the software assets…throughout all stages of their lifecycle.*[v]

ITIL's definition of SAM is a bit broader, and focuses on additional facets of infrastructure and configuration management, which is to be expected. Between these two resources, we can create or enhance our existing comprehensive view of our software assets.

Information Technology Asset Management

Broadening our scope even further, we begin to enter the domain of Information Technology Asset Management, or ITAM. ITAM is a practice that addresses the management of both software and hardware assets. The International Association of Information Technology Asset Managers (IAITAM) was formed to bring together a professional community of ITAM practitioners to bring further structure to the ITAM discipline. From their perspective, they offer the following:

IT Asset Management (ITAM) is a set of business practices that incorporates IT assets across the business units within the organization. IT Asset Management joins the financial, inventory and contractual responsibilities to manage the overall life cycle including tactical and strategic decision making. Across every organization type and size, ITAM delivers service to everyone in the organization to facilitate current and future business operations. The mission of the IT Asset Manager is to maximize benefits while minimizing risks from IT assets in their organization.[vi]

TREND: *"TechNavio's analysts forecast the Global IT Asset Management Software market to grow at a CAGR of 9.4 percent over the period 2012-2016. One of the key factors contributing to this market growth is the increased need for ITAM cost reduction. The Global IT Asset Management Software market has also been witnessing the emergence of cloud computing. However, the high deployment cost could pose a challenge to the growth of this market."*[vii]

Research MOZ (2010)

Asset Repository

As the organization of the asset inventory begins to take shape, it is highly recommended that a robust asset repository be established; one that is designed to house the asset details as well as the critical asset's metadata, such as availability, functionality, re-use potential, re-use assessment, tags (structured, social, collaborative folksonomy, etc.) and key relationships. Please note that an Asset Repository is separate and distinct from the traditional Configuration Management Database (CMDB) that most IT professionals are familiar with. Think of them as being complementary to one another, based on common identifiers between the two data stores that provide referential integrity. Consider the contrast in the following table below:

Asset Repository	Configuration Management Database (CMDB)
An Asset Repository maintains details about each asset such as the information associated with asset discovery, inventory management, contract management and financial management. This repository is concerned with all assets, regardless of their status and purpose, and the financial cost and legal compliance associated with each asset in the infrastructure. An asset repository is likely to be significantly larger than a CMDB in most organizations.	A CMDB is a special-purpose repository of Configuration Items (CIs) that contains business service to IT service relationships plus the underlying dependencies of related assets. A CMDB is populated only with CIs containing business IT service relationships and inter-dependencies that are linked to critical business services. The CMDB purposely excludes non-service related assets in order to maintain focus on the mission of service management.

From *IT Asset Management - A Cornerstone for Accelerating ITIL Success* [viii]

The key take away from getting organized is the importance of establishing and maintaining a true and accurate inventory of technology assets deployed across the enterprise to the level of granularity that is appropriate for the organization. Successful firms house and manage that inventory in a centralized asset repository that includes automated data quality mechanisms in place. Controls should be in place to ensure that no rogue or unidentified technology assets are introduced into the ecosystem. Without this level of rigor, the firm must accept the risks inherent with partial information. Imagine trying to manage your personal financial portfolio without knowing all of the assets in it!

ORGANIZE
OPTIMIZE
CATEGORIZE

Step 2: Categorize

Getting all of our assets organized into a common repository is a huge but critical undertaking when it comes to managing a portfolio of technology assets. If the common repository is not considered the authoritative source of record, results will be marginal at best as analysis of the asset metadata based on the repository will always be suspect. Once doubt is cast on its accuracy, the repository becomes a tool of convenience, used when it supports one's position, and ignored when it does not. For the sake of our discussion here, we'll assume that the efforts to organize and catalog the firm's technology assets were successful, and that the repository has been deemed to be a reliable source of information.

Many of the Software Asset Management and Information Technology Asset Management efforts focus on quantitative analysis, providing asset counts and corresponding asset identifiers such as type (hardware vs. software) and platform (Linux vs. Windows). The challenge, however, is to fill the gap when it comes to taking a more qualitative capability view of the technology asset portfolio. In other words, now that we know what assets we have at our disposal, we need to understand more about what those assets do and the value that those assets bring to the organization. That understanding can be enhanced by categorizing our assets in relationship to our defined Enterprise Architecture Building Blocks and to Business and Technology Capabilities.

Building Blocks

Several Enterprise Architecture frameworks support the concept of Building Blocks, which generally describe discrete elements or components as being either 'hard' (hardware and software combined), 'soft' (software only) or 'connector' (block-to-block facilitators, such as interprocess communication channels) types[ix]. The folks at Sun Microsystems suggested the use of a list of building block properties several years ago as a means of assessing the building block's non-functional capabilities. I have taken that list and adapted it based on industry experiences in the delivery of Service Oriented Architecture (SOA) solutions over the years since the list was initially published in 2004[x].

- Scalability — Ability to replicate the building block multiple times to scale the level of service it provides (horizontal scalability) as well as take advantage of larger processing capacity (vertical scalability)
- Functional Isolation — Ability of the building block to support multiple applications in a service-oriented, context-agnostic fashion
- Configurability vs. Customization — Ability of the building block to be used in a variety of adaptive ways through configuration rather than through one-off customizations (or extensive contextual IF-THEN-ELSE logic)
- Coarse-Grain/Fine-Grain Reusability — Ability of the building blocks to provide modular services and configurable process orchestration (compound, singular, hybrid)
- Portability — Ability of the building block to be ported to and deployed across multiple operating platforms, including both OS and delivery channels
- Integration — Ability of the building block to support service requests through a common Application Programming Interface (API) and published Software Development Kit (SDK)
- Asynchronous Communication —Ability of the building block to support asynchronous messaging between other building blocks

TOGAF Building Blocks

The Open Group Architecture Framework (TOGAF) addresses the concept of building blocks as part of their defined Architecture Content Framework (TOGAF 9 Part IV). You'll see by the quote below that they are taking a slightly abstracted view of the building block concept, which is useful for our purposes as well:

A building block's boundary and specification should be loosely coupled to its implementation; i.e., it should be possible to realize a building block in several different ways without impacting the boundary or specification of the building block. The way in which assets and capabilities are assembled into building blocks will vary widely between individual architectures. Every organization must decide for itself what arrangement of building blocks works best for it. A good choice of building blocks can lead to improvements in legacy system integration, interoperability, and flexibility in the creation of new systems and applications.[xi]

TOGAF further refines building blocks into Architectural Building Blocks (ABBs) and Solution Building Blocks (SBBs).

	Architecture Building Blocks	Solution Building Blocks
Characteristics	• Capture architecture requirements • Direct and guide the development of SBBs	• Define what products and components will implement the functionality • Define the implementation • Fulfill business requirements • Are product or vendor-aware
Specification Content	• Fundamental functionality and attributes • Interfaces • Interoperability and relationship with other building blocks • Dependent building blocks with required functionality and named user interfaces • Map to business / organizational entities and policies	• Specific functionality and attributes • Interfaces • Required SBBs • Mapping from the SBBs to the IT topology and operational policies • Specifications of attributes shared across the environment • Performance, configurability • Design drivers and constraints • Relationships between SBBs and ABBs

TOGAF Building Block Characteristics and Specifications [xii]

TOGAF describes how the Building Blocks are to be introduced/ extended/refactored during the Technical Architecture Phase and then further incorporated during the Opportunities and Solutions Phase as part of their Architecture Development Methodology (ADM). The topic of TOGAF Building Blocks is worthy of its own white paper, so we'll assume for this discussion that the organization has already integrated some form of Building Block methodology and lifecycle management process into their current solution delivery process.

Our reason for discussing Building Blocks here in terms of asset categorization is to ensure that the institution's defined Building Blocks are linked to the discrete entries in the asset repository either as a relationship (preferred) or as metadata about the asset. This allows us to map realized assets against their relevant architecturally defined Building Blocks to quickly see both gaps and overlaps.

Capabilities

In previous writings, I've introduced and discussed the concepts of asset segregation across four discrete dimensions:

- Business Capabilities
  Areas of competency required by the organization to achieve its vision and strategy; What the organization must be good at
- Business Components
  Non-Technical resources and operational assets available to support the execution / delivery of the Business Capabilities; How the organization is structured/operates
- Technology Capabilities
  Systemic features and functions required by Business Components to deliver Business Capabilities; What the systems must be able to do

- Technology Components
  Technical resources and systemic assets that provide Technology Capabilities; What systems are available

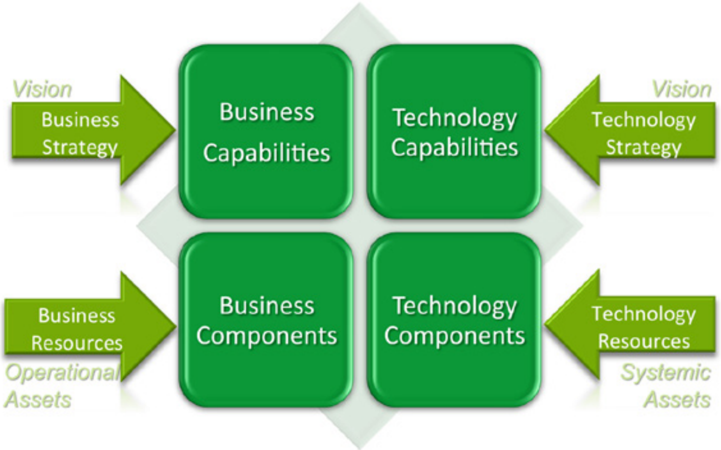

Figure 3: Business and Technology Asset Categorization [xiii]

Just as we associated various Building Blocks to discrete assets in the asset repository, we should also tag those assets with their corresponding Business Capabilities, Business Components, Technology Capabilities and Technology Components.

Enterprise Reference Architecture Alignment

Another important way to categorize assets within the repository is to identify how the assets relate to the organization's defined Enterprise Reference Architecture. There are many definitions for 'Reference Architecture' in use across the industry, and our goal here is not to promote one over the other. I offered the following working definition in an earlier work:

Reference Architecture is a term that describes a discretely articulated set of constructs, or building blocks, that define particular functional and non-functional domains relevant to the entity. An organization's Enterprise Reference Architecture Model is simply the collection of published Reference Architectures used to govern the generation of strategically aligned solutions.

The Reference Architecture defines the organization's de facto architectural standards to be applied when delivering strategic capabilities within a particular discipline or domain.[xiv]

Reference Architecture Domains

Most organizations roughly segregate their Reference Architecture domains into three segments; Functional, Non-Functional and Utility. Functional Reference Architecture domains usually define a preferred tech stack aligned to a specific business function, such as payments of offer management. Non-Functional Reference Architecture domains typically define the organization's chosen platforms such as operating systems or enterprise service messaging systems. Utility Reference Architecture domains are often a hybrid of functional and non-functional capabilities assembled as a commodity service, such as a document imaging platform, or security entitlement and authentication services. Hopefully there is consistency across the domains so that the Functional Reference Architecture specifications are based on approved Non-Functional and Utility Reference Architecture components.

Part of the categorization exercise is to determine under which Reference Architecture's jurisdiction the asset falls. Just as a citizen will be subject to federal, state and local guidelines, assets in the asset repository will likely fall under more than one Reference Architecture domain. Many organizations find it useful to select a single 'primary' domain, and additional 'secondary' domains as relevant. Other organizations struggle with the concept of a forced fit to a single over-arching domain and want to have several primary domains. My advice is to avoid turning this into an extended and overly academic debate – you'll hit the point of diminishing returns pretty quickly. Work to move the participants to consensus on the level of granularity, and recognize that as the data gathering progresses, course-corrections will become more obvious.

Core Strategic Asset Classification

Although all assets will be aligned to their appropriate Reference Architecture domains, that doesn't imply that they are actually part of the defined Reference Architecture. It will be important to identify whether or not the asset is considered to be a 'core strategic' asset; meaning it is either explicitly identified as part of one or more Reference Architecture definitions, or it is fully aligned. This becomes an important facilitator to the Enterprise Reference Architecture roadmap process. Those assets which are not aligned can quickly be identified and a deprecation strategy established, based on the impact analysis that the Asset Repository can facilitate.

Once the correlations have been identified and added to the asset repository, we are left with a very rich information set that can provide a comprehensive view of our entire portfolio; both from a quantitative view (how many assets we have) and a qualitative view (what kind of assets we have and what they do). Below is a sample series of data-gathering questions we could use for the categorization process:

- Is this asset instance scalable?
- Is the asset functionally isolated?
- What is the asset's level of configurability?
- What level of asset reuse granularity is supported?
- Does the asset have a published API and corresponding SDK?
- Does the asset support Asynchronous Processing?
- Does the asset represent a defined Building Block? If so, what type, and what are the asset's characteristics and specifications?
- What Business Capabilities does the asset support?
- Which Business Components consume the asset?
- Which Technical Capabilities does the asset provide?
- Is the asset aligned to its prevailing reference architecture? If so, which one?

- Does the asset conform to governing compliance regulations? If so, which one(s)?
- What does is cost to operate the asset?
- What is the book value of the asset (i.e. is it being amortized and would removal accelerate cost recognition)?

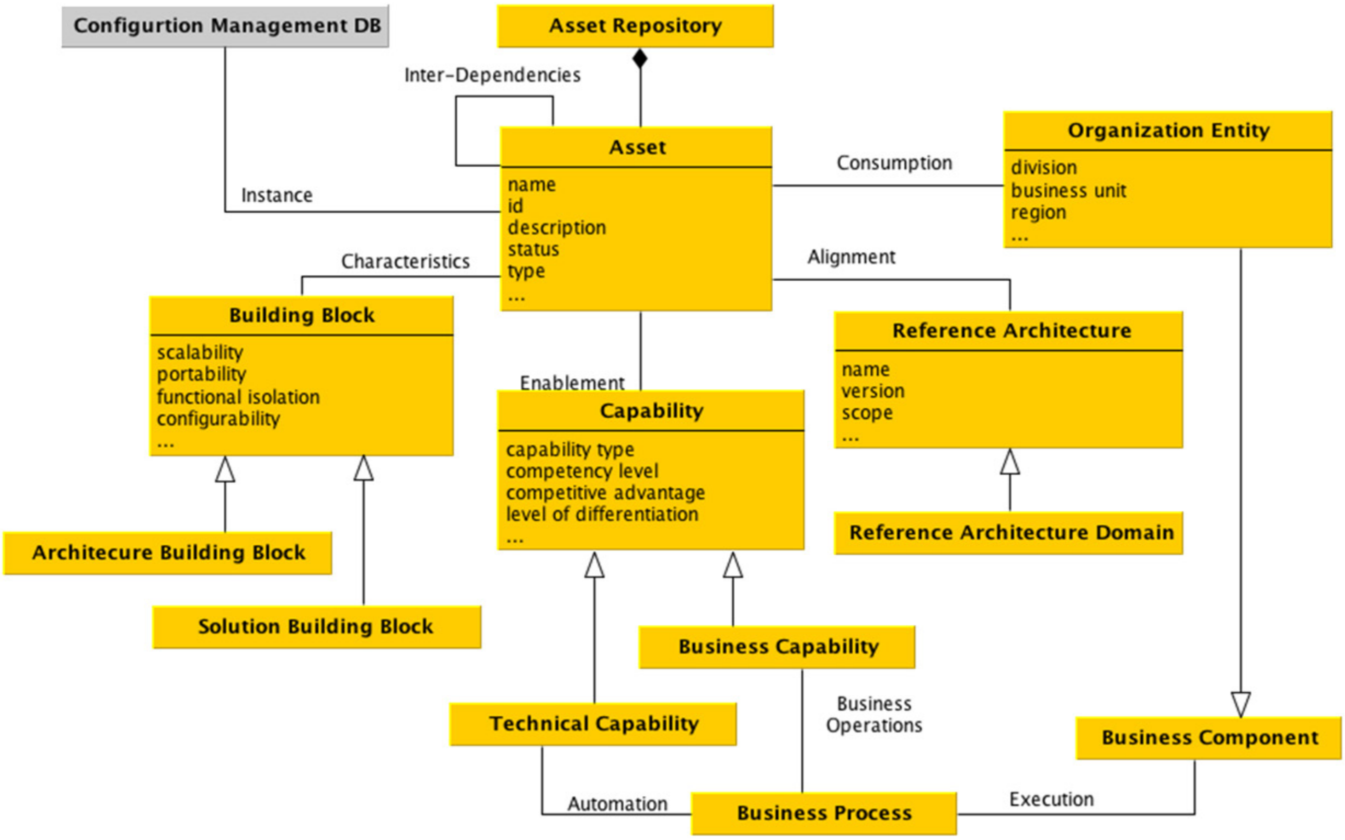

Figure 4: Sample High-Level Asset Repository Categorization Information Model

As you look through the list above, some aspects or attributes will likely resonate more with you and your organization than others. This list is not intended to be exhaustive or prescriptive, but rather designed to help ignite the thought process. An important aspect when establishing the categories you'll ascribe to your assets is to make sure that each attribute is bounded by a discrete set of values, either by enumeration or by association. Capturing open-ended data points may be interesting, but will be very difficult to analyze.

To help visualize how this all fits together, Figure 4 provides a basic information model using the various asset categorization schemas we've discussed so far. Again, consider this as more of an extendable logical view from which to pattern your asset repository information model, rather than a precise model upon which to base the asset repository.

As the asset repository takes shape, analytic reports can be run across various dimensions and criteria in order to identify where there are gaps in functionality or redundant capabilities across the asset portfolio. This all sets the stage for comparing our asset repository against our reference architecture roadmaps and driving asset portfolio optimization.

TIP: *Treat the Asset Repository like any other critical business information repository. Take the time to create a thorough conceptual Platform Independent Information Model view of the entire enterprise asset domain, capturing and validating relevant classes, attributes, constraints and relationships before data gathering and categorization begins. Otherwise the asset repository will quickly become unmanageable and yield only limited value.*

This also becomes a very powerful tool for tracking Enterprise Reference Architecture roadmap progress and conformance. Adding an asset consumption model also helps identify early adopters of strategic solutions as well as those parts of the organization which are struggling moving off of non-strategic platforms.

ORGANIZE
OPTIMIZE
CATEGORIZE

Step 3: Optimize

Now that we've inventoried our assets and categorized them in meaningful ways, we are at the point where we can begin to optimize our assets through basic asset management techniques. Figure 5 below shows a typical asset management lifecycle, which begins with an evaluation of the needs of the situation (investment opportunity). Assuming there is value in moving forward, a decision is then made to either utilize an existing strategic asset (as is or through extension) or to obtain/create/deploy/utilize a new asset. Once the asset has been placed into service, it must be reviewed on a regular basis to ensure that it is still adding value to the organization and that it has not introduced any new risks. For those assets which are no longer adding value, the asset disposition process begins, following a re-evaluation of the needs to determine if the asset's functionality is still required. If not, the asset lifecycle simply comes to a natural conclusion and unnecessary business processes are discontinued.

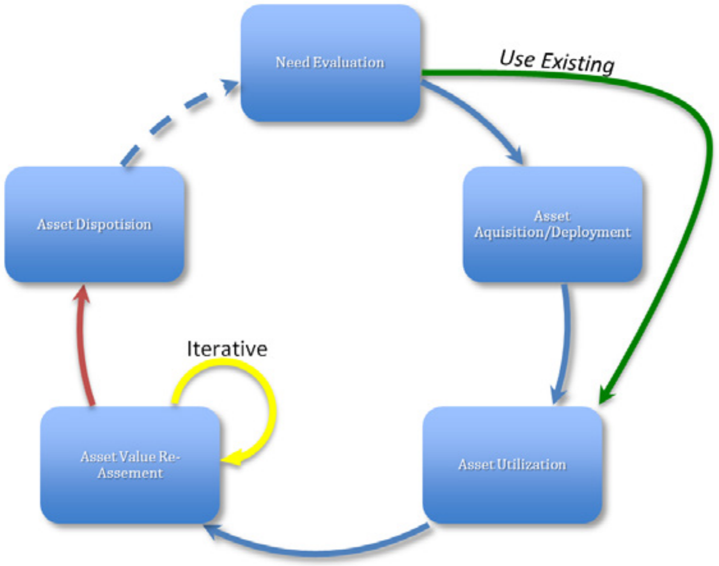

Figure 5: Typical Asset Management Lifecycle

If the need is still legitimate, then the same reuse vs. buy vs. build process begins again. It may seem counterintuitive to remove an asset whose functionality is still required by the organization. Bear in mind that the Asset Value Re-Assessment is not to challenge the need for the functionality, but rather to ensure that the asset itself still represents the optimal solution. This is where the combination of capabilities and Building Block non-functional attributes begins to help shape an informed decision. For example, we can project an asset's scalability to meet projected capability demand based on its Building Block characteristics identified in the prior step.

Asset Portfolio Management

Putting the asset repository together was an important step, but unless we begin to manage it with the same proactive invest/divest point of view that comes from an asset portfolio management discipline, all we have is a large inventory of assets. Asset management is all about maximizing value and controlling risks across the portfolio. Our goal is to exploit the use of strategic, high-value assets across the organization, limit and control the deployment of assets of only marginal or moderate value, and reduce or eliminate the use of low or negative value assets. To help identify the value group of the assets, we ascribe them with a simple asset investment categorization of Buy, Hold or Sell.

Buy

Assets that are assigned an Asset Investment Category rating of Buy are considered to be strategic to the operation of the organization and are fully aligned with the appropriate reference architecture domain(s). These assets provide a high level of value to the organization in terms of their impact on the firm relative to their cost of operation and ownership. Many organizations find basic ROI (Return on Investment) to be a measurable assessment of the asset's value. Be careful when establishing a financial valuation model; they can be very helpful only if they are reasonable and can be applied consistently across a heterogeneous collection of assets.

Note that the value of a business process is not the same as the value of the asset that supports the business process. If users of an archaic system are still somehow able to complete a high-value process, that doesn't mean the asset it self has high value; it means the capability is of high value, and the supporting systemic assets should be optimized.

If you find that a large portion of your Enterprise Reference Architecture assets do not qualify as being given an investment category of Buy, it is time to take a deeper look at the components within the reference architecture stack. Don't assume that just because the asset has been linked to the reference architecture in the past that it automatically should be granted a Buy rating. The asset valuation assessment process should be the driver of the rating.

Hold

Assets that are assigned an Asset Investment Category rating of Hold are still providing a respectable level of value to the organization and should be maintained in their current state. Non-discretionary investments should continue to be made in the asset, such as maintaining regulatory compliance levels, performance tuning and so forth. Discretionary investments in Hold assets, however, should go through a very rigorous and challenging cost/benefit analysis, with aggressive payback period requirements. If a strong business case can be made, then the investment should be considered. Once the investment has been made, a re-examination of the asset's investment categorization should be made as well.

To be clear, fighting requested investments in Hold assets will not be popular. Quite the contrary, many proponents of the investment will typically become quite defensive of the asset and lobby for its continual expansion even though its classification indicates that it does not represent a significant value to the organization. Continually focus on the business case, challenge the numbers presented, and don't hesitate to invest when the financials support moving forward.

That being said, as a rule it should be difficult to get approval to make discretionary investments in Hold assets – difficult yes, but not impossible when presented with a legitimate and compelling business case. If firm fiscal discipline is not going to be employed, then there's no point to differentiating between Buy and Hold assets.

Sell

Assets that are assigned an Asset Investment Category rating of Sell have outlived their useful economic life and are providing low or even negative value to the organization. These assets represent a financial and operational drag on the organization, as time and money that could be diverted to more valuable assets gets wasted running and maintaining these under-performing assets. Only minimal non-discretionary investments should be made in Sell assets, with no funding made available for discretionary changes, such as new functionality or enhanced user interfaces. Aggressive conversion and retirement plans should be put in motion to take Sell assets out of the portfolio as soon as feasibly possible.

The asset's prevailing Enterprise Reference Architecture definitions come in to play by providing the definitive list of go to alternatives per their respective roadmaps. Those Sell assets that are still required but do not have a clearly identified replacement highlight gaps in the Enterprise Reference Architecture that need to be addressed. Moving to a Buy replacement is optimal; moving to a Hold replacement is acceptable if a suitable Buy asset is not available. However, moving from one Sell asset to another is seldom a good alternative, as the investment could be better used on higher performing assets in the portfolio.

Finally, don't assume that all Sell assets are older legacy assets or outdated platforms. While its true that this is often the case, firms that engage in a high degree of industry-leading research and development may find their asset repository littered with Sell assets that have been neglected over the years and continue to draw resources.

Putting Buy-Hold-Sell to Work

Getting all of the assets in the asset repository properly assigned their appropriate investment category won’t matter much if there is no change in organizational behavior. Early architectural deliverables such as Architecture Contracts, Architecture Definition Documents and Architecture Principles from TOGAF (or similar Enterprise Architecture Framework artifacts) can be used to guide downstream development efforts in the adoption of Buy and Hold assets. Architectural peer reviews can be used to challenge architects who continue to propagate Sell assets in their solution sets before the discussion leaves the architecture community. Performance measurements can be established based on how well the architecture community is driving conformance towards Buy assets and the number of gaps identified in the Reference Architecture when Sell alternatives are lacking.

Design reviews with the development community should target and reject solutions based on Sell assets. A fair but firm appeals process should be in place to provide an opportunity for appellates to make a case for investing in non-strategic, non-performing assets. Financial disincentives can be used as well, such as not allowing non-strategic asset development costs to be amortized or forced to recognize a higher technology chargeback rate to establish an ‘asset retirement fund’ to offset eventual asset disposal costs.

TIP: *Enforcing rigorous Asset Portfolio Management is not easy, and certainly not for the faint of heart. People will vigorously defend a system they claim to ‘hate’ when they believe it is at risk of being taken away from them.*

Turn the conversation around to focus on why the organization is moving to higher-performing assets. Track savings based on the retirement of under-performing assets and demonstrate how the process is freeing up capital for investment in higher-performing assets. Present a compelling business case rather than a compelling technology case.

Project portfolio and pipeline prioritization processes can be enhanced by weighting requests based on their primary investment category. For example, projects that are made up of mostly Buy assets would be given preference over comparable projects made up of mostly Hold assets. Discretionary investment requests against Sell assets can be quickly rejected. Consumption metrics by Asset Investment Category can easily be obtained from the asset repository to identify ‘worst offenders’ and target phased asset retirement strategies across the organization.

Recommended Reading

ITIL V3 Guide to Software Asset Management Book
Office of Government Commerce (2009)

Reducing Risk and Maximizing Investment Through IT Asset Management: A Practitioner's Perspective
Allen, Doyle, Lehr and Fisher (2011)

Reusable Asset Specification
OMG (2005)

Making Enterprise Information Management Work for Business
John Ladley (2010)

Conclusion

There is a lot of work required in building a comprehensive asset repository. Being able to optimize that repository requires and extensive categorical assessment of the assets across multiple dimensions. Optimization of the asset portfolio from an asset management perspective requires an understanding of which assets to invest further in, which assets to maintain in a steady state, and which assets to eliminate.

The benefits of establishing an Asset Investment Classification mechanism include:

- Alignment of asset investments to the Enterprise Reference Architecture roadmap;
- Assessment of the Architecture Community's ability to influence investment into strategic assets across the organization;
- Establishment of a design conformance mechanism;
- Acceleration of asset retirements, and
- Objective means of determining how discretionary asset investment should occur.

References

[i] Covey, Stephen R. (1989). The 7 Habits of Highly Effective People. P. 95. New York, NY: Simon and Schuster

[ii] International Organization for Standardization. (2006). International Standard ISO/IEC 19770-1 Information Technology – Software Asset Management. P. v. Geneva, CH: International Organization for Standardization

[iii] International Organization for Standardization. (2013). Standards Catalog – ISO/IEC JCT 1/SC 7 Software and Systems Engineering. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=45086. Accessed July 14, 2013

[iv] 19770.org (2013). SAM Standards Working Group (ISO/IEC JTC1 SC7 WG21) 19770-1. http://www.19770.org/projects/19770-1/. Accessed July 1, 2013

[v] Rudd, Colin. (2009). ITIL's Guide to Software Asset Management.

[vi] International Association of Information Technology Asset Managers, Inc. (2012). What is IT Asset Management?. P. 2. Canton, OH, USA: IAIATM, Inc.

[vii] Research MOZ (Online Research Reports). http://www.scribd.com/doc/138653123/Global-IT-Asset-Management-Software-Market-Trends-2012-2016. Accessed July 6, 2013

[viii] Davis, Jason and Chip Galusha (2007). IT Asset Management — A Cornerstone for Accelerating ITIL® Success. P. 3. New York, NY, USA: CA

[ix] Radhakrishnan, Ramesh and Rakesh Radhakrishnan. (2004). IT Infrastructure Architecture Building Blocks. Page 2. Santa Clara, CA, USA: Sun Microsystems

[x] Radhakrishnan et al, P. 5.

[xi] The Open Group. (2009). TOGAF 9 – The Open Group Architecture Framework. P. 500.

[xii] TOGAF 9. P. 501-502.

[xiii] Sereff, Guy B. (2012). Launching an Enterprise Business Architecture Practice: A Playbook for Getting Started. Figure 5. Franklin, TN, USA: Guy B. Sereff

[xiv] Sereff, Guy. (2013). Information Security Integration within the Enterprise Reference Architecture Model - Part 1: Foundation. P. 5. London, England: Orbus Software.

Orbus Software
3rd Floor
111 Buckingham Palace Road
London
SW1W 0SR
United Kingdom

+44 (0) 870 991 1851
enquiries@orbussoftware.com
www.orbussoftware.com