It's rare that a business stays just as it began year after year. So if we agree that businesses evolve, it follows that information systems must evolve to keep pace. So far, so good. The disconnect occurs when an enterprise's management knows that the information systems must evolve, but keeps patching and whipping the legacy systems to meet one more requirement. If you put the problem in its simplest terms, management has a choice about how it will grow its information systems. If there is a clear strategic vision for the enterprise, it seems logical to have an equally broad vision for the systems that support that strategy. Managers can thus choose to plan evolution, or they can react when reality hits and “evolve” parts of the information system according to the latest crisis.

It's a bit of a no-brainer as to which is the better choice. But it's also easy to understand why few enterprises pick it. Conceiving, planning, and monitoring systems that support a long-range strategic vision is not trivial.

Enterprise-wide information systems typically start from a base of legacy systems. And not just any legacy systems. They are typically unwieldy systems of systems with a staggering array of hardware, software, design strategies, and implementation platforms. To make the job even more difficult, “enterprise-wide” often means city to city, state to state, or even country to country.

Getting these pieces to seamlessly interact and evolve according to long-range strategic business objectives may seem like mission impossible; for a large distributed organization, however, it is mission critical.

**BLUEPRINTS FOR A SHARED VISION**

In our 20 years of working with businesses and government agencies, we’ve seen a common misperception in tackling this problem: Most enterprises start at the detailed architectural design, failing to develop what we call an enterprise information technology architecture (EITA).

The EITA is the blueprint for creating enterprise-wide information systems, and as such, describes a set of information system architectures. There are lots of reasons to have an EITA. Because it is enterprise-wide, everyone communicates from the same perspective and has a shared context for deciding about IT investments. You have a way to describe new information systems or strategies for modernizing existing ones. You have a clear path for developing enterprise information systems and a starting point for detailed design and construction.
Figure 1. A framework to build an enterprise information technology architecture (EITA) for the US Dept. of the Treasury.

The picture starts with the enterprise’s mission, business requirements, and IT objectives.

The vision directly drives the architectural principles and the business view, which starts the process of elaborating and structuring the business vision.

Standards and a technical reference model make sure everyone has a common understanding of functions and terms.

The result is the information needed to implement a set of information systems, including the schedule and budget.

Iteration among multiple views helps represent different stakeholders and gauge the effects of change.

The vision directly drives the architectural principles and the business view, which starts the process of elaborating and structuring the business vision.

Standards and a technical reference model make sure everyone has a common understanding of functions and terms.

The result is the information needed to implement a set of information systems, including the schedule and budget.
Building the EITA is not the highest view in the big picture, however. To be able to maintain and evolve it intelligently, you must first create an EITA framework or adapt an existing one.

**CREATING A FRAMEWORK**

A framework is a conceptual model for developing an EITA. It is not an architecture per se. Rather, it presents a set of models, principles, services, approaches, standards, design concepts, components, and configurations that guide the development of specific architectures. You may already be familiar with conceptual models such as the Zachman Framework, Technical Architecture Framework for Information Management (TAFIM), and the Open Group Architectural Framework (TOGAF).

Creating a framework for an enterprise may be as simple as tweaking an existing framework or as complicated as inventing your own. In most cases, you won’t have to start completely from scratch. For example, although we developed a framework specifically for the US Department of the Treasury, we drew concepts from the Zachman Framework, TAFIM, and TOGAF.

Even if you decide to adapt an existing framework, you still have a fair amount of work. You’ll need to customize it to suit your organizational culture and vocabulary, for example. You’ll also need to put the framework through several dry runs, which will inevitably generate some lessons learned. Be prepared to spend some time refining and adding more details to the new framework because you won’t get it right the first time.

A first step in creating the framework is to carefully evaluate and understand your business environment. You can’t possibly decide on the framework structure without this. For example, the US Treasury, like most major organizations, is large and highly decentralized with rapidly changing business requirements. Thus, it has to be able to develop enterprise applications consistently and in keeping with the pace of its business mission.

The framework we created (Figure 1) begins with a business vision—including the IT vision—which determines IT goals and objectives. Together, the business and IT visions drive the business view and architecture principles. With this flow, we were able to ensure responsiveness to the Treasury’s business objectives.

A key challenge is to make sure the framework guides overall architectural design but is still broad enough to withstand all the modifications from different groups within the enterprise who will need more specific support. Development groups are notorious for customizing modeling approaches (object or function models, for example) to represent particular entities and services.

To provide the structure and guidelines for EITA development, most frameworks will include a set of architectural principles, architectural views, a technical reference model, and a standards profile.
CAPTURING THE BUSINESS VISION

After you evaluate the business environment, you need to start figuring out how to capture it in the framework. The architectural principles and technical reference model (Figure 1) are the foundation for this.

For example, in TA FIM, as well as in our framework, architecture principles are simple, direct statements of how an enterprise wants to use IT. These statements establish a context for architecture design decisions by translating business criteria into language and specifications that technology managers can understand and use. A architecture principles put boundaries around decisions about system architecture.

Guiding principles are critical to any architecture framework because they provide a consistent, shared vision for developing new architectures. Architects, managers, maintenance staff, and so on will use these principles to make sure development initiatives are in line with the enterprise's overall strategic vision. Without principles, the architects may build technically perfect systems but not ones that meet the needs of the enterprise.

A architecture principles evolve as the overall business mission evolves, often slowly. When an architectural principle doesn't seem valid anymore, consider rewriting or replacing it. However, although principles are driven by business requirements, they are not necessarily tied to technology changes, so don't rewrite a principle for that reason alone.

Also remember that developing a set of architectural principles is a real balancing act. We've seen organizations develop architectural principles that were really just detailed implementation rules and they ended up hamstringing the architects. State principles in a way that provides solid direction, yet gives architects enough freedom to be creative in their solutions.

In our Treasury framework, for example, we have only 20 architectural principles organized by view. The principles state the direction, but say nothing about how to get there. Each principle has from five to 10 implications that give a better idea of how that principle will affect business operations and what actions might be required. A gain, however, there is no mention of any particular implementation path.

Here are three architecture principles from the work view:

1. The Department will provide access to information to authorized users to perform their jobs independent of physical location.
2. Information should be captured in computer-readable form as close to the source of origin as possible, including external sources and forms prepared and submitted by the public.
3. Common user interface components and standards should be used to provide user interface services.

An implication of 2 and 3 is that the Department will attempt to provide some common access point for retrieving, filling out, and submitting electronic forms to Treasury agencies. And an implication of 1 might be that the Department should create a corporate information directory that identifies the sources, types, locations, sensitivity, and owners of each data element.
Implications are valuable because they help define specific components of an EITA. In the Treasury example, a component could be a corporate information directory, a communication infrastructure among the business locations of the organization, or a data format that lets data be shared across information systems.

PROVIDING A COMMON UNDERSTANDING

When developing an enterprise architecture, everyone has to speak the same technical language and be on the same page in understanding what the architecture is supposed to do. In a distributed organization like the Treasury, this can be a real challenge.

A technical reference model (TRM) describes the main parts of a complete information system as a set of services (Figure 1). Because it categorizes the services by functional area and doesn’t care how they’re implemented, you get a standard information system model that guides the design and development of customized information systems. However, customization often means resolving issues that affect interoperability, portability, and scalability, so if everyone has a different idea of the system being discussed, you could end up with a real mess.

The standards profile (Figure 1)—brief descriptions of available standards—helps to provide a common language. It also helps promote interoperability, portability, flexibility, and scalability because it serves as a reminder to adhere to certain standards or as a suggestion to use standards to avoid unnecessary work. By following open-system standards, for example, you can provide common services and create reusable building blocks. These, in turn, mean faster, cheaper, better systems. As the demand for interoperability increases, standards are no longer a luxury. They are mandatory.

CAPTURING MULTIPLE PERSPECTIVES

An EITA attempts to portray a complex reality, and thus is difficult to represent with one model or view. Just as we look at a city plan in different ways, so stakeholders have different perspectives or views of an EITA. A database administrator will be interested in information entities. A business analyst will be interested in the business processes and functions. A network engineer will be interested in the network technical infrastructure and topology. Multiple views help manage complexity, separate concerns, and address the different life spans of the architecture’s elements. These views also help you understand how individual information systems fit into the larger structure that supports the organization’s total business operations.

Most architectural frameworks acknowledge the need for views, although they tend to disagree on what those views are and how many you should have.

In the simplest terms, an architectural view is a well-defined vantage point that describes a specific set of concerns. Each view is modeled by a set of elements, relationships between elements, and constraints on elements and relationships.

For the Treasury framework, we included five views (Figure 1)—business, work, function, information, infrastructure—but every organization is different. The views you choose must reflect how the information systems are used without regard to how an individual system will be built. They must capture the business processes and functions, the information that must be captured and manipulated, the physical distribution and organization of the customers and workforce, and the technical infrastructure.

The work, function, and information views in our framework collectively describe the Treasury’s business operations as defined by its business view. These views are mission-specific and organization-dependent. The infrastructure view, on the other hand, reflects the services and environments to support the information, function, and work views. Therefore, it should not be mission- and organization-dependent.

In defining views, take care to maintain consistency among them. A good way to do this is to map entries between views. For example, by mapping information entries in the information view to functions in the function view, you can more easily see how some change in function will affect the data that function needs. A nother example is mapping functions to organizational elements in the work view, which will tell you the best location for particular functions, such as customer support.
How Enterprise Architecture Frameworks Compare

Athough enterprise architecting is a relatively new discipline, several frameworks are already available to help you develop an enterprise information technology architecture. Most consist of the framework elements we described in the main article, as well as an EITA development approach, but they vary in the type of guidance and specific views. The matrix is by no means a complete description. Here, we just hit the main points.

<table>
<thead>
<tr>
<th>Framework</th>
<th>Description</th>
<th>Noteworthy components</th>
<th>Architectural views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zachman Framework</td>
<td>* Based on practices in traditional architecture and engineering * Provides a basic structure for creating and maintaining architectural representations of an organization</td>
<td>* 2D matrix of views representing the architecture * Set of rules for applying the framework</td>
<td>* Data, function, network, people, time, motivation * Five divisions per view: enterprise model, system model, technology model, components, and functioning system</td>
</tr>
<tr>
<td>The Open Group Architectural Framework (TOGAF)</td>
<td>* Guides the evolution of an organization’s technical infrastructure * Same generic technical reference model as TAFM and IEEE Posix.0</td>
<td>* Technical reference model * Standards information base * Architecture development method</td>
<td>* Function, implementation, physical * Three views above decompose to more detailed views</td>
</tr>
<tr>
<td>Technical Architecture Framework for Information Management (TAFIM)</td>
<td>* Guides the evolution of the DoD technical infrastructure * Based on IEEE Posix.0 * Same technical reference model as TOGAF and IEEE Posix.0</td>
<td>* Technical reference model * Adapted information technology standards and guidelines for developing more detailed standards profiles * Guidance for designing architectures using TAFM life-cycle management process</td>
<td>* Computing, data management, communications, security</td>
</tr>
<tr>
<td>CIO Council Federal Conceptual Model</td>
<td>* Aids in managing the development and maintenance of a federal enterprise architecture and in related decision making * Partitions architectures into levels and provides a set of layers for modeling, similar in structure to the Zachman Framework</td>
<td>* Set of architecture principles * Emphasis on segmented or incremental architecture development * Description of how an architecture based on the framework would be organized</td>
<td>* Data, systems, infrastructure * Each view has five user perspectives: planner’s, owner’s, designer’s, builder’s, and subcontractor’s</td>
</tr>
</tbody>
</table>

**Business view**

The business view is the why of information systems, describing the business processes and information that support the business mission and operations. Without this view, you could not understand how the organization operates and could not align information systems to support business operations. The business view is often divided along major organizational elements, such as finance, human resources, manufacturing, and research and development. A gain, if you fail to consider the business’s goals, objectives, and processes, you may develop a great architecture for the wrong business. A though this seems obvious, we’ve seen enough people fail to do it. EITA development must involve the business users or you pretty much waste the architectural effort.

**Work view**

The work view describes the who and where of information systems, specifying how organizational components should be distributed to business locations and how locations should communicate and coordinate resources. It also describes the business operations each area performs and the divisions and types of work the components should do. Ultimately, the work view will help you decide where to locate information systems to support business operations because it contains descriptions of the overall organizational hierarchy.
Function view

The function view defines the how of information systems, describing the common information flows, business functions, and processes that manipulate and manage business information to support business operations. It also describes the logical dependencies and relationships among business functions.

The function view is often described in terms of two kinds of core functions. The first is the functions and processes common across organizational units. The second is the functions specific to or specialized for individual units. By identifying common functions, you can build physically distributed information systems that can be developed, used, and maintained centrally. You can also reuse common frameworks and code, and cost-effectively support business processes. By identifying specialized functions, you can specify customized versions of common systems or develop unique information systems to meet the particular needs of one business unit.

Information view

The information view defines the what of information systems, specifying the major information entities needed to operate the business, their relationships, and how they map to business processes, units, and locations. Without the information view, you may find it hard to understand what information is shared and where, or to determine the best way to distribute that information across functions, the infrastructure, or physical locations.

Infrastructure view

The infrastructure view defines the how to of information systems, describing the supporting services, computing platforms, and internal and external interfaces the information system needs to run. It also describes the connectivity for these platforms that enables the interoperability for applications and information. Because the infrastructure view describes the technology you need to meet the business requirements, it helps guide the other architecture views by making it clear that the technology exists to implement them.

DEVELOPING THE ARCHITECTURE

Once you have the framework in place, you’ll need a plan for building the EITA. The framework defines the architecture, but it can’t do all the work. A well-thought-out, documented process helps you avoid missing important steps and ensures that you develop the EITA consistently and in a way that others can understand.

The process

The process has four main steps.

Define a shared IT vision. This typically means establishing a committee to draft the vision and architecture principles. Committee members should be upper-tier functional and IT managers, but you’ll also need a working group of midtier functional managers and IT personnel to plan, work on, and maintain architectures. Remember that architecting is an open process. Don’t hide the team away and stamp everything “Confidential.” Invite participation and circulate drafts for review and discussion.

Figure out where you are now. Establish a technology baseline and a business view that will become the starting point for architecture development. Be sure to establish buy-in by involving teams of architects, planners, and managers directly in the creation and review of deliverables.

Decide where you want to go. Establish the target architecture by examining technology trends and defining various views and standards profiles. Don’t get bogged down if information is not available. Just make your best guess and document any assumptions. Keep things at a fairly high level by focusing on architecture decisions and save some creative work for the designers.

You should put “incremental and iterative” in huge type somewhere to remind everyone that these are key philosophies in creating an enterprise architecture.

Form a plan to get there. Outline specific actions for moving the enterprise from its current state to the desired target in the most cost-effective way. Develop a modernization strategy, establish the groundwork for migrating existing information systems, and identify any new information systems you need. We’ve found the schedule for moving to the target architecture is critical. The team’s morale suffers if you don’t show results early on. Set schedules such that deliverables arrive within weeks, not months.

These development steps are by no means sequential. This is a huge undertaking, and you will go back and forth a lot before it’s over. In fact, you should put “incremental and iterative” in huge type somewhere to remind everyone that these are key philosophies in creating an EITA.

It’s impossible to specify an enterprise-wide architecture in a single effort. Technology and business conditions change so rapidly that the architecture would be out of date before it’s complete. The people we’ve seen attempt the “big bang” to architectural development either run out of time and money or have requirements change midstream.

It makes more sense to evolve the architecture by focusing on high-level issues and developing more detail as the need arises. This has two advantages. First, the high-level view is more digestible and provides insight to key decision makers. Second, because your focus is on both critical
and near-term requirements, you get results quickly. You can use input from, say, business process reengineering, to update the architecture without doing more analysis.

We can’t say enough about the need for training and mentoring architects who will be building the EITA framework for the first time. Don’t just throw the framework document at them. Actively gather lessons they have learned and use them to later revise and elaborate the framework.

Finally, architecting is a process, not an event. You don’t want to produce a shelf document that lets individuals go back to their ad hoc ways of making IT decisions. Establish a procedure for updating and regularly reviewing architectures and updating them when needed.

The people

You will need to select and train enterprise architects with multiple skill sets according to Eberhardt Rechtin, a system architect has three essential skills: human, technical, and conceptual. System architects must be able to construct a technically coherent and complete vision of what the system should look like, and what it should do. Conceptualization is the essence of architecting. Selecting a product or technology family is usually up to individual IS project managers, but the architect may have to define corporate standards. System architects must be able to identify emerging technologies, be aware of cutting-edge technologies, and be able to specify the technologies needed to develop or evolve the organization’s system architecture. Finally, the architect must be able to communicate the system architecture to the organization and work well with both implementers and users.

Ensuring a consistent, coherent vision to direct the evolution of the enterprise’s information systems is difficult enough with a development framework. Without one, it’s impossible. The concepts we’ve outlined are just a task list in the work to establish an enterprise architecture, but they give you some idea of the breadth of the job ahead of you. In upcoming issues of IT Pro, we’ll walk through the how-to of enterprise architecting, covering architecture modeling approaches, what makes a quality architecture, steps in the architecture development process, and the specific challenges of architecting for a distributed organization. Meanwhile, we hope we’ve given you a start toward a more realistic vision of your enterprise architecture.

Frank J. Armour is an associate research professor of information technology at George Mason University. Contact him at farmour@gmu.edu.

Stephen H. Kaisler is the director of systems architecture at the Office of the Sergeant of Arms, US Senate. Contact him at Steve_Kaisler@saa.senate.gov.

Simon Y. Liu is an assistant director of information management and security staff at the US Department of Justice. Contact him at Simon.Y.Liu@usdoj.gov.