Encyclopedia of Information Systems and Technology

Phillip A. Laplante, Vladimir Stantchev, Jeff Voas, Anne-Marie Smith, Qiuming Zhu, Morris Chang, Irena Bojanova, David Alan Grier, Jia Zhang, Thomas Costello, San Murugesan, Jim Goldman, Joanna DeFranco

Enterprise Architecture: Planning

Publication details
Kirk Hausman
Published online on: 29 Dec 2015

How to cite :-
Accessed on: 18 Aug 2023

PLEASE SCROLL DOWN FOR DOCUMENT

Full terms and conditions of use: https://www.routledgehandbooks.com/legal-notices/terms

This Document PDF may be used for research, teaching and private study purposes. Any substantial or systematic reproductions, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The publisher shall not be liable for an loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
Enterprise Architecture: Planning

Kirk Hausman
Assistant Commandant, Texas A&M University, College Station, Texas, U.S.A.

Abstract
This entry provides an examination of elements to consider in enterprise architecture. It also identifies measures of success and common pitfalls, reviews the architectural roles and requirements, and provides considerations for formal governance and framework adoption.

Before a building can be constructed, a detailed layout of all the structural, environmental, decorative, and functional elements must first be compiled. This detailed layout, called a blueprint, ensures that adequate resources and room to expand are present and that all elements will work together in the final product. The blueprint is reviewed many times before construction begins to ensure that elements from many different disciplines will work together well—that electrical and plumbing fixtures are properly placed, that the windows and doors produce the desired effect, and that the materials specified are suitable to the environment.

Before an enterprise network can be constructed, or further developed from its existing state, a design similar to a physical construction blueprint must be created. Instead of building codes and construction methodologies, this design must identify the guidelines and strategies that will form the basis for the network architecture, along with standards and frameworks that will be used to guide its creation.

This entry focuses on the function and traits of an enterprise architect, information technology (IT) governance principles, and several architectural frameworks that can be used as templates for developing an enterprise network architectural plan.

BEYOND PLATFORM SELECTION

Many attempts at enterprise architectural planning revolve around selection of a particular technology as the standard that will be applied to purchasing choices. Such a “platform architecture” might include listings of approved hardware platforms, operating systems, desktop configurations, office productivity suites, content management and database management systems (DBMS), programming languages and development tool suites, collaboration platforms, directory and identity management solutions, and even mobile technology solutions approved for staff use or purchase reimbursement.

Architects of commercial platforms may select a particular vendor stack, such as Microsoft or IBM, for enterprise-wide standardization in order to ease integration and planning. Open-systems architects may select only a set of standards around which all purchases must be made, with minor platform or application variation allowed at the user or facility level if deemed worth the added complexity. Regardless of the particular approach chosen, selecting standardized platform architectures can provide direct benefits that are easily conveyed to stakeholders:

- Economies of scale can be gained through aggregated purchasing mechanisms spanning multiple business units that formerly purchased technologies in an ad-hoc manner using locally directed funding. Enterprise-wide licensing and per-processor licensing models can often save tremendous amounts of money compared to individual per-user licensing throughout a medium-to-large organization.
- Applications are more easily integrated and tested, depending on the level of commonality provided through platform specification. The more elements of the platform architecture are shared, the greater is the ease in initial implementation and later additions of functionality. Selecting a standard or vendor with a large body of third-party offerings allows an organization to readily adopt newly emergent technology offerings as they reach viability.
- Users and technical support staff can be more efficient throughout the enterprise, as they can gain a greater working knowledge of a smaller number of disparate solutions. This can reduce both staffing and training requirements, and allow greater mobility within an extended organization to meet marketplace agility needs. When all applications are standardized, a user trained in the standard suite of productivity
applications will require very little training if he or she is transferred to another department.

- Selecting widely used platform architectures provides a greater body of community support when issues arise or when new technologies are being implemented. When problems or questions arise, it is easier to find effective solutions if the platform architecture is in common use and enjoys strong community involvement in implementation and development.
- Control is simplified because only the initial selection process must include all possible competing alternatives. After the platform architecture has been identified and detailed, selection needs only to evaluate alternatives within the identified platform’s offerings.

Enterprise architecture would be greatly simplified if platform selection was its only feature. Within small, well-funded organizations, many of these “low-hanging fruit” may be plucked readily as value-for-effort. However, selection of a single platform specification to be implemented across a medium-to-large organization will rapidly encounter many stumbling blocks, often including the following:

- Many packaged applications rely on a particular set of technologies for operation. Legacy equipment such as automation and equipment control systems may rely on an embedded control technology, while other applications may only operate against a particular type of relational DBMS such as an Oracle or MySQL instance. Changing these solutions to meet newly-identified platform architectural guidelines can be problematic, requiring time, effort, and expense in acquiring alternatives that fit within the identified platform—assuming that such an alternative exists. Application virtualization systems are often tied to the fundamental architecture of the network itself, relying on the authentication and access control system to control application package assignment, deployment, and updating. The architect can address this issue by carefully identifying all such dependencies before beginning enterprise reconfiguration actions, so as to avoid mid-upgrade loss or instability of key services that can stall or terminate enterprise-wide projects.
- Replacement of existing, working systems may be beyond the budgeting ability of many organizations and business units. While there are times when a functional but expensive enterprise solution may be tossed out in favor of an alternative, the majority of businesses must enact incremental evolutionary change rather than outright revolutionary change simply because the cost of replacing everything outweighs the benefits of immediate standardization. An architect’s knowledge and understanding of existing revenue streams and any constraints on stream allocations are critical to understanding and successfully planning the scope and pace of enterprise-transforming projects.
- Changes in technology standards, development of new offerings, and the constant evolution of network threats can often outrun an established platform specification. By the time an organization can transition its technologies to meet the new standard, several new versions of an identified platform option might have been developed to add new features or close vulnerabilities not identified at the time of the original assessment. This is particularly true when dealing with governmental agencies, educational systems, and organizations with complex bureaucracies, because of the lag between specification and purchase. An identified best-in-breed at the time of original product evaluation might be woefully inadequate compared to later entries into the same application space by the time the new fiscal year rolls around and funds are available for acquisition. The architect must constantly review emerging technologies and keep abreast of changes in common technology devices, configurations, and uses.
- The scale of need and expertise may vary widely within an organization, creating a requirement for different levels of operational capability that may not be reflected by a one-size-fits-all platform specification. A commonly encountered version of this involves the dominant enterprise Microsoft platform, which includes both desktop (Access) and server (Structured Query Language Server) relational database (RDBS) solutions. Both solutions use a similar interface for query/view design and share many elements in common; however, the workgroup-scale Access application is not suited to the same level of concurrent use as its more robust Structured Query Language Server counterpart. Conversely, training to support each RDBS application differs greatly at the level of expertise required, because the workgroup product lacks many of the index, trigger, stored procedures, and other features necessary in its enterprise counterpart. Proper project management practices during the planning phase will aid in achieving an effective transformation by identifying existing solutions that are in place and compiling skills and responsibility matrices to identify available resources, both human and technological, as well as any shortcomings that must be met by acquisitions or training.
- Partner organizations may not follow the same standards, creating difficulties when the platform architecture must interoperate with that of another organization whose decisions are made along a differing standard or cycle of upgrade. This problem becomes more apparent as organizations are increasingly coupled with partners for service provision,
regulatory mandate, or business opportunity. Poorly selected platform architecture or the choice of little-used technology standards can create barriers to business integration and lose opportunities for the organization well beyond simple issues of whether the solution meets an originally identified specification. This problem arises often during mergers and acquisitions when platform architecture decisions have followed different courses in the originally separate organizations. As an example, elements of the U.S. Department of Homeland Security experienced difficulties with e-mail between organizational elements when the department was created from previously separate functional organizations. Platform differences caused communications difficulties between some divisions of the new organization, even though they were all using the same underlying standard (SMTP) for e-mail. Use of common platforms and applications in architectural redesign will reduce the likelihood of poor interoperability with partner organizations or external services such as hosting or cloud computing providers. Customization of operating system, application, and service configurations beyond accepted industry standards should be kept to a bare minimum to protect both interoperability and upgrade potential.

- Individual business units or key stakeholders may be unwilling to accept the platform architectural specification. A wide range of technology preferences may factor into the local mandates provided by business-unit IT decision makers. In addition to preference, individual business units may lack the expertise and skills necessary to implement a newly specified platform standard—particularly one that differs considerably from the platform architecture currently in use. Matrices and surveys of existing resources compiled during the planning phase will help identify requirements for training to upgrade support capabilities. The platform architect should also ensure that support leadership is included in the technology council that provides change management controls, to ensure that issues are presented, considered, and addressed before initiating a change.

- Implementation of alternative standards or platform specifications without a review of service impact or analysis can compromise service availability and user capabilities, and may create push-back from user and management-level consumers in addition to technical support teams. This type of issue is common when movement to an older standard platform is necessary due to a merger between organizations and the main organization is using software versions that are older than the ones in the newly acquired organizational element. Users who have become accustomed to features and functionalities provided by the newer software may experience significantly reduced efficiency due to a lack of certain features. Users may also feel “cheated” by being forced to roll back to the “outdated” application platform or software standard.

Certainly, these are not the only barriers to adopting a platform specification, but they illustrate clearly why the practice of enterprise architecture must take into account elements that extend beyond the range of platform specification alone. Beyond specification of a set of standards or vendor products through which the enterprise will operate, the enterprise architect must also consider integration and communications requirements, application-specific requirements, available and emergent communications protocols, business integration mandates, legislative and regulatory compliance constraints, scale and scope requirements, event management and alerting, reporting mandates, storage and architectural requirements, identity management mandates, and a wealth of other aspects.

Without guidance and consideration of these elements both in their existing form and following emergent trends, the enterprise architect may make short-term decisions that fail to provide longer-term efficiencies—in fact, some decisions can even create greater cost burdens if improperly applied. Architecture is more about plotting a course through a variety of choices than about making a particular application selection. Technical frameworks and standards must be balanced against the strategic goals of the business while minimizing the negative impact of any changes on the end user. Rather than being a goal, technology must always remain a means to facilitate user requirements.

In the end, a project that produces the desired change in technology but negatively impacts user and client service remains a failure. According to the Project Management Institute, roughly two-thirds of all projects fail. To avoid expending unnecessary effort on failure, the enterprise architect must know when to initiate a project, when to decide the time or resources are not right for a particular change, and when to call a halt to avoid spending resources on a failed effort. Always remember that sunk costs (money already spent) should not be used to determine whether to continue a failing or troubled project.

WHERE LIES SUCCESS?

Successful enterprise architecture must convey some form of advantage to the organization or it serves no purpose. While it might once have been possible for an organization to gain a competitive edge by using e-mail for internal communication or a website for user contact, these solutions are used by such a broad range of organizations that they create no operational advantage.
Although doing without them might have a negative effect, their presence alone no longer constitutes a particular competitive advantage. Enterprise architecture must provide additional value to the organization.

A few qualities that may indicate successful enterprise architecture include:

- Providing a clear upgrade path to technological progress
- Defining standards for interoperability with existing and emergent technologies
- Minimizing undesirable redundancy and optimizing fault tolerance
- Reducing risk and enhancing continuity of operations
- Reducing support and operational costs
- Improving operational opportunities for interaction and mobility
- Recognizing architecture as a valuable contribution to the enterprise

It may at first seem strange that one indicator of a successful enterprise architecture is that the architecture is recognized as valuable. However, architectural choices can have a profound effect on an organization’s capabilities and opportunities for growth. An architecture that is completely in the background, which is never noticed by the users and stakeholders, may be too readily discarded as an unnecessary effort or expense during organizational changes or economic cutbacks. It is important for long-term viability that not only should good decisions be made, but that they must also be communicated to implementers and users. Successes, failures, and even near-misses should also be conveyed to primary stakeholders so that the value of the architecture remains known, identifiable, and justifiable within the overall business plan.

An enterprise architect may be aided by other architects, a change management council, or an integration competency center to ensure that all decisions that are made fit within all necessary technologies and operational mandates. However, it is the architect’s vision of the enterprise that guides all policies, standards, guidelines, and specifications that will be enacted within this scope. The enterprise architect must be able to convey the value of this vision to both stakeholders and implementers, and must be able to see far enough ahead so that decisions made allow changes when necessary.

As with any project, the project manager (enterprise architect) must be selected as early in the process as possible and must be given the authority to negotiate for necessary resources and designate standards for architectural changes. There can be only one person in the role of chief architect: Effective responsibility cannot be assigned to a group.

In smaller organizations, the enterprise architect may also be the chief technologist or lead developer. This can be acceptable in a small venue, as the architect may be able to learn and consider all possible aspects of the organizational technology needs. In larger organizations, it is important to distinguish enterprise architecture from development and IT implementation. While nontechnical managerial staff members cannot be effective enterprise architects because they lack the necessary understanding of service interrelations, even effective IT professionals may be poor enterprise architects if:

- **Their personal preferences for technology specification do not provide the best opportunities for their organization as a whole.** Many organizations attempt to bypass the tendencies by placing a nontechnical individual in the chief information officer (CIO)/chief architect’s position. While this may avoid the problem of IT preference, it creates a situation in which the primary decision maker responsible for architectural vision must rely primarily on the advice of others—a situation akin to sheep asking wolves to advise them. This also leads to project selection in isolation, where a change in collaboration platform might be decided on without considering the impact on related services such as an organizational intranet portal or a customer management system that leverages the collaboration platform for critical functionality.

- **Their vision lacks clarity and understanding of the needs of the organization, or changes too often to provide a useful direction.** Nothing can doom an enterprise architectural effort more effectively than a leader who does not know where the organization is going. Fuzzy goals tend to lead to unremarkable results that lack metrics for assessing success. Like a boat without a tiller, an architect without direction tends to lead nowhere. Worse, in some ways, is the architect whose vision changes with each new month’s trade journals or based on which stakeholders have most recently argued their needs. An inconstant leader can cause implementers to waste valuable time jumping from project to project, at times working only to undo what has just been partially completed because the architect has just heard about the newest process-of-the-month. Nothing illustrates more how lost a leader is than such wandering in the wilderness of alternatives.

- **They are unable to convey their vision to stakeholders, implementers, or business consumers.** The most knowledgeable technologists, skilled in every aspect of their craft and instilled with the most incredibly clear vision of successful enterprise architecture, are useless if they cannot effectively convey their vision and its value to others. Many technologists eschew the “soft skills” necessary for project management, negotiation, and selling the idea of their vision and its worth to the organization. For those to
whom technology itself becomes the goal, conveying its value beyond existence appears unnecessary. This form of tunnel vision can doom an enterprise architectural effort to failure. The most beautiful painting in the world has no value if it is kept in the dark—so too with an architectural vision.

- **They fail to remain knowledgeable about technologies in use and in development.** While the CIO/chief architect should focus on aspects of technology that extend beyond the day-to-day implementation details, it is impossible to command the respect of information technologists charged with implementing the vision without maintaining a thorough knowledge of existing and emerging technology trends. Not only may architects miss potential opportunities, they may lose the support and input from individuals whose expertise may be the key to greater success. Buy-in from implementers within the technology service arena is often developed only in the presence of expertise on the part of the architect. While this role need not have deep expertise in all areas, the architect should be an expert in at least one area and conversant in the functions of all others in order to best gain the trust and support of other technologists. Knowledgeable architects can also bridge disparate skills as a force multiplier, producing an outcome greater than that possible by team members acting apart.

- **They fail to lead.** The worst of all possible architects are those who are so busy gathering requirements, evaluating options, negotiating with their peers, reporting to their superiors, and getting feedback from stakeholders that they never do anything. It is easy to slip into a routine so filled with the process of developing architectural guidelines that nothing is ever produced as a result. The pursuit of perfection is seductive and pointless—what must be accomplished is a solution that is good enough for business needs and good enough to allow the next generation of technologies to be managed in turn.

Many different means can be used to measure the state of architectural guidance and management within an organization. Table 1 provides a simplified maturity model based on those offered by Gartner Research. Far too many organizations operate at the lowest levels of maturity, where decisions on architecture and implementation are missing altogether or function in chaos merely as reactions to the latest problem or need. Proactive effort and cost is generally far lower than reactive, because reactive actions include resources necessary to undo what has been done in addition to resources necessary to implement the new course of action. Without a higher level of planning, coordination, and vision, these organizations cannot take advantage of the true potential of their IT resources.

### THE ARCHITECT

If a successful architecture can produce advantages, then what makes a good architect? This question is tied closely to the job the CIO/chief architect must perform in order to effectively identify, strategize, and resolve organizational needs and technology drivers that affect the enterprise. The architect is responsible for many tasks, such as:

- **Identifying data and its movement.** The enterprise architect must identify data that is being managed and maintained throughout the enterprise, along with the paths through which it is transferred, archived, or eliminated in order to properly plan for its continuity under a formal architectural style. Many times, the identification and elimination of undesirable redundancy in data stores can produce strong benefits from direct cost savings in hardware, software, and support to indirect cost savings through security of sensitive or protected data.

- **Defining technical architectural guidelines.** Standards, technology selections, protocol selections, guidelines for identity management, update management, security and recoverability statutes, computer use policy specifications, and all other aspects of the technology Architecture must be coordinated by the enterprise architect in the overall vision.

- **Integrating existing resources.** The enterprise architect cannot, in almost all cases, simply throw away...
everything that exists and replace it with new solutions. Embedded systems, legacy equipment, merged business units, and partner relationships may all bring different technology solutions into the enterprise. The enterprise architect must plan for the inclusion of these elements into the strategy.

- **Communicating the vision.** The enterprise architect must convey the benefits of the vision to stakeholders, the details to the implementers, and the benefits and purpose to the users. At the same time, this communication cannot be one-way. The enterprise architect must constantly be accepting useful input to be included in the evolution of the vision to encompass emerging needs and solutions.

- **Improve quality.** The results of the enterprise architect’s efforts must provide value to the organization, improving the quality of IT operations. Architects whose solutions simply replace existing systems with a different version of the same thing may find it difficult to justify the continued expense without some measure of value or quality improvement. Careful metrics identified before and after each phase can aid in the identification of the value provided, requiring the architect to practice more than simple technology selection in order to be a success.

- **Program management.** Strong project and program management skills are vital to ensure that all phases of architectural reconstruction are handled in an efficient manner using well-defined strategies. In addition to maintaining technology operations, which many former IT managers are more than capable of continuing, the enterprise architect must manage numerous projects simultaneously to control costs and ensure that activities on the critical path are addressed ahead of activities with flex remaining. It is far too easy to seek “easy” projects to illustrate ongoing success to sponsors, but this can lead to missed deadlines and cost overruns when those easy projects are not ones that must be completed first to maintain overall project and program timelines.

**PROVIDING A PLAN**

The chief architect must be able to identify the purpose of IT solutions, align them with business requirements, and communicate their value to both technical implementers as well as nontechnical stakeholders. Subordinate IT professionals at times may want to try out new solutions that have no applicability to the business environment—distinguishing new business opportunities from expensive toys is a valuable skill for the chief architect. The chief architect must be willing to simply say “no” to nonproductive projects and inefficient ways of doing business. The chief enterprise architect is rarely a popular person, as entrenched professionals with established professional and personal standing can be very influential when complaining about changes that are made counter to local preferences.

The chief architect must always be looking to identifying crises that may emerge as a result of new network threats or emerging technologies. Disaster recovery and business continuity may hinge on effective IT planning, in which good decisions and an eye on potential large-scale disruption may keep the business on track while competitors fail. Concerns about global pandemics, terrorist activities, and natural disasters all add to the scope of an enterprise architect’s planning, because his decisions may determine whether an organization survives or simply ceases to exist in the worst case.

**COMMUNICATING THE VISION**

Chief architects must be able to see the big picture but paint it on small canvases. Translating the vision into simple, concise pieces is vital both in educating stakeholders and conveying directives to those responsible for executing planned changes. A solid business case for each primary element, along with strong project management skills that can bridge multiple projects into a cohesive program within a common framework, can aid in rapidly identifying a process gone awry or one that is being ineffectively implemented. Unhappy implementers can often be identified here, where their lack of enthusiasm may be adversely affecting project success and timeliness.

Chief architects must be as at home talking to primary stakeholders or individual users as they are comfortable communicating in the language of business or technology. They must be able to effectively convey elements of the vision at all levels of the business, but also be able to dig down to the individual details of the application of the vision so that communication can flow both ways. Ultimately, they need to be able to sell the idea to stakeholders, convey it to implementers, and document everything so that metrics can be measured.
against the process of change. Like all project managers, enterprise architects may spend up to 90% of their time communicating to implementers, change management functionaries, stakeholders, project sponsors, peers, and individual users.

**BALANCING VALUE AND RISK**

The chief architect must be able to identify value in the present architecture, as well as identify missing elements that need to be filled. Many times, some elements of efficiency can be acquired rapidly. These “lowhanging fruit” are easily obtained, and it is important to identify these for initial implementation in order to show value—but care must be taken to ensure that jumping too quickly does not incur undesirable opportunity costs, excluding opportunities for greater improvements later. Establishing clear business goals and balancing them against selected technology solutions is not a simple task, as each choice may close off other alternatives later.

Chief architects must identify metrics and goals for success and failure, while also planning and identifying risks that will arise from each step toward these goals. Cost and risk factors and various constraints must be considered during each planning phase, both at the immediate per-project level and across the entire enterprise planning process. Adding vulnerability or exposing protected data during a transitional phase may not be legally allowable under regulatory and operational mandates, while relying on a technology that is still in the incubator stage may produce an unacceptable level of risk to proofing guidelines and purchasing requirements. Security and regulatory compliance must always be considered in strategic planning to ensure that assets are not exposed to risk during transition.

**The Business Architect**

Organizations that employ complex application solutions or ones with public-facing business applications such as Web shopping carts or business intelligence portals may find it necessary to assign a business architectural role. Whether combined with the chief architect’s position or implemented separately, the person holding this operational role must understand and translate all business strategies and processes into requirements that can be addressed through technology selection or development.

The business architect’s role may include technology planning for business-to-business, business-to-consumer, partner integration, service-oriented architecture selection, and management of heterogeneous or legacy application suites that require data gateway translation or information transfer for operation.

**The Technology Architect**

The technology architect role becomes necessary in organizations that employ a wide range of technological solutions, or in which application development or customization is used extensively. This role requires deep technical experience, often in one or more programming disciplines, and the technology architect acts to ensure that application development and modification are performed within the strategies detailed from the chief architect’s vision.

The technology architect is responsible for guiding application design style selection, such as service-oriented, scrum, or waterfall-type development, as well as testing new techniques and technologies for potential use within the extended enterprise. Though often considered the most fun job, this role of the technology architect is not just to try out all of the new technology toys, but to be able to draw a hard line when a solution reaches end of life or is determined as falling outside the organization’s needs.

**The Lead Architect**

In support of a chief architect, a lead architect may participate in senior management forums, serve on the change management council, or lead implementation groups in applying the strategies developed from the chief architect’s vision. In small to medium enterprises, these roles are typically combined. A lead architect will often lead forums, integration competency center reviews, requirements-gathering initiatives, and other similar tasks necessary to identifying business and technology elements for integration into the overall architectural plan. This role is particularly valuable when attempting to integrate multiple cross-enterprise initiatives during mergers.

**Outsourced Architecture**

Some aspects of enterprise architecture can be outsourced to external expertise. In general, this is done in order to gain access to skills that are not present or not present in sufficient depth within the existing human resources of the organization. When making a strategic change in storage architecture or federated identity management, outsourcing the planning roles can be effective while internal personnel are being provided the necessary training to understand a new solution’s implications. During mergers and acquisitions, for instance, an outside expert may be able to begin the migration process while local IT resources are retrained into the parent organization’s architecture.
Outsourcing the developing of a strategic guide or blueprint may provide a means to save time within limited internal enterprise architect resources, or when internal political issues provide the need for an outside viewpoint to settle on one solution as the common thread for development of a strategic vision. Documentation of existing resources can also be outsourced in order to save time and to reduce the impact of internal business-unit politics on discovery and resource identification. Critical functions such as enterprise architectural change or security may be outsourced in some cases, but responsibility for services, regulatory compliance, and data protection ultimately remains with the organization.

One additional area that lends itself well to outsourcing is that of testing and compliance review. Testing new technologies for a fit within the enterprise architectural vision may be best performed by a third party in order to ensure that findings reflect technology interactions and not simply local preferences. Similarly, compliance audit and review should be performed by a dedicated or external agency in order to ensure that internal bias or simple familiarity do not cause the review to overlook areas of concern. Audit functions must not be conducted solely from within the IT organization, as it is impossible to obtain a fair and thorough report of variance when the individual reporting an issue is subordinate to the functions or personnel being assessed.

**Multiple Architects**

Small to medium-size enterprise architecture can easily be coordinated by a single chief architect, with additional supporting focused architects added to cover capacity shortfalls as the enterprise scope expands. This can produce an optimized architecture, capable of the most complete synthesis of homogeneous network coordination with the greatest possible reduction in data and network complexity and cost. As enterprise networks increase in size or span multiple business units with independent purchasing capabilities, the load may be distributed across multiple architects’ purview, provided the organization has matured to some form of matrixed structure. A single chief architect or chief technology officer must still be identified in all cases, because someone must ultimately make choices between alternatives and be responsible for success or failure of the effort.

Distributed federated architecture for very large enterprises can be configured serially or in parallel, depending on the nature of the enterprise and its goals, as noted in Fig. 1. Serial architecture divides individual aspects of the overall architectural vision so that each element is guided by its own chief architect. Communication among the chief architectural roles creates a community environment within the coordinated guidelines dictated by each architect’s contributions. This is similar to a university or corporate environment in which each business unit operates within a negotiated general vision but maintains some aspects of localized control.

Parallel architecture is more like a city, in which no common vision is implemented and each silo makes decisions for access within its own boundaries. Multiple autonomous heterogeneous silos of parallel operation yield localized responsiveness at the cost of opportunities for efficiency, much like physical city planning efforts. Parallel architectural control scenarios tend to experience large undesirable redundancies in data, hardware, and staffing requirements and do not adapt to wide-scale upgrade as well as more closely coordinated architectural forms. This is the least efficient format for large enterprise networks, as differing architectural decisions may generate standards conflict, compromise security, and create large areas of overlapping expense and operational effort.

**CREATING A SYMPHONY**

Federated architectural solutions, whether distributed serially or in parallel, must have a central chief architect to set basic policy and provide the highest-level vision—all other forms of federated architecture will produce conflicting internal elements and impair long-term efficiency and viability. This need is seen in many other operational arenas: Cooking, music, education, corporate control, and military strategy all rely on coordination under a designated leader in order to avoid chaos.

Without a master chef, the sous-chefs and all others working in a commercial kitchen might produce a
variety of very nice dishes but would be very unlikely to create an integrated masterpiece of culinary art. A general officer in the military may rely heavily on the support of senior staff officers, but in the end must make decisions alone so that an army can move toward a single purpose. The federal Sarbanes-Oxley legislation formalized responsibilities built into the framework of corporate governance, mandating specific attention and control in the corporate sector. A master conductor must work to bring together the disparate instruments present so that an orchestra can produce a symphony—one that will differ from the same music played by the same orchestra under a different conductor’s baton. Each of these scenarios represents the same need found in enterprise architecture—someone, ultimately, must hold the baton.

Governance

The art of enterprise architecture relies on similar high-level coordination to gain advantages in agility, cost reduction, and operational efficiency. The symphony that can result from effective coordinated enterprise governance depends on a clear vision, strong leadership coupled with executive buy-in and support, and an effective means of communication with those responsible for their own areas of expertise. Some assessments detail the process of enterprise architecture as follows:

1. Creation—Identification of the business drivers and requirements that create a need for enterprise coordination. This may be a simple need for cost-effective technology utilization, or made more complex by regulatory mandates and partner intercommunication requirements.

2. Discovery—Identification of the individual protocols and technologies that come together to form the executive-level vision that will guide technological development, purchase, and organization.

3. Implementation—Enacting changes, developing policies, communicating requirements necessary to implement the vision in actual terms.

4. Governance—Overseeing and managing the process that guides technology decisions, implementation actions, and all other decisions that fall within the guidelines of the technology enterprise.

However, governance is more fundamental than first appears in this process. Enterprise architecture translates business requirements into technology planning that must include strategic and operational decisions. These decisions must in turn contain decision making for capacity, cost, recovery, and survivability that must be more than theoretical abstracts. Governance is more about communicating between strategic roles (Chief executive officer, chief financial officer, CIO, business-unit leaders), operational roles (managers, partner representatives, regulatory agents), and infrastructural roles (integration competency centers, IT implementers, training staff).

Without some mechanism for governance, lines of communication and authority can become hopelessly tangled and doom the architectural effort before it has even begun. Many formal systems for IT governance exist, including:

- **Information Technology Infrastructure Library (ITIL).** Perhaps the most widely adopted standard for enterprise governance, the ITIL is a best-practice set of guidelines for operational control. Because of the level of detail in the ITIL, it can produce revolutionary change—with all of the benefits and costs that it entails. Without strong management commitment, care in training and specifying each ITIL process, and planning for a 3–5 year implementation effort, this methodology can be a bit hard to handle. As a living document, the ITIL continues to evolve to meet new challenges that follow emergent technology standards and options and is widely used in large enterprises such as governmental and multinational corporations.

- **Control Objectives for Information Technology (CobIT).** This detailed governance model produced by the Information Systems Audit and Control Association (ISACA) grew from an audit and control methodology and is also widely recognized, with a strong community supporting its continued evolution. Its sponsoring agency, together with the IT Governance Institute, produces a large number of focused-compliance guides to help CobIT apply more specifically to individual guidelines that may apply to specific industries and business sectors (Sarbanes-Oxley, COSO) and to specific technologies (zOS, Linux, SAP).

- **International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 27002.** The British governmental 7799 standard was adopted by the International Standards Organization as ISO 17799, which enjoys wide use throughout many business sectors—although it is often found combined with other governance methodologies due to the large number of translation guides that exist to bind this standard’s elements with those of the other methodologies. The ISO/IEC 27002 standard has since replaced the older ISO 17799 standard, reflecting changes and more recently emergent requirements.

Many other formal systems exist for IT governance, and any sufficiently detailed system with a strong community of support and regular updates to its standards...
could benefit the architectural process. Control objects from the ITIL, CobIT, ISO 17799, and other similar standards are aligned in many different studies, allowing organizations to use elements of those that fit best. It is vital that enterprise architects include an understanding of IT governance in their planning and vision. Few organizations of significant size will find that a single governance or control standard meets all possible needs, so methodologies are often layered at different operational levels (as shown in Fig. 2).

Architectural Models

In addition to an understanding of governance methodologies, the chief architect should also have an understanding of formal architectural frameworks that can be used as fundamental models during the architectural process. These frameworks can provide extraordinary resource, although many carry an inherent bias toward specific technological viewpoints and should be considered in that light. Formal enterprise architectural frameworks include options such as the following:

- **POSIX 1003.23.** One of the earliest formal frameworks for enterprise architecture, this standard was developed by the Institute of Electrical and Electronics Engineers as one of several standards related to software compatibility for Unix operating system variants. Because of its origins, this standard is coupled closely with Unix and Unix-like platform environments.

- **The Zachman Framework.** A widely-accepted benchmark model, this framework was developed by the Zachman Institute in order to model enterprise architecture in terms of scope, business model, system model, technology model, and other details when measured against the standard questions asked: what, how, where, who, when, and why. As a high-level model, the Zachman Framework provides an excellent starting point for architectural theory.

- **The Open Group Architecture Format (TOGAF).** TOGAF has been developed by the Open Group, a consortium of information system vendors, software suppliers, and user organizations. It is a very detailed, extensive living document that can be used as a starting point for developing commercial platform enterprise models, though its origins couple the document itself more closely to the open systems methodology.

- **The Federal Enterprise Architecture Framework.** This framework emerged from a series of policies and guidance efforts within the U.S. federal government, aimed at providing a standard framework for planning interoperable and sustainable computing. It derives details from earlier governmental standards, including the Office of Management and Budget’s EA Assessment Framework, the General Accounting Office’s IT Framework, the Treasury Enterprise Architecture Framework, the Generalized Enterprise Reference Architecture Model, the Performance Reference Model, the Data and Information Reference Model, the Application-Capability Reference Model, the Technical Reference Model, and other similar frameworks. It is detailed, extensive, and continues to be updated to reflect the changing needs of governmental IT enterprises.

- **The Gartner Enterprise Architecture Framework.** Developed by Gartner Research, this framework attempts to provide a comprehensive model for the development of both commercial and open-source enterprise environments. It is divided into parts, with half of the model addressing technology, while the other half addresses business strategy and process. Although it remains less developed in terms of a body of reference guides and compliance checklists, this framework enjoys steady growth in the body of documentation and research provided by its parent organization.

- **The North American State CIO Enterprise Architecture (NASCIO).** An emerging leader in enterprise architectural planning, this model is developed by the NASCIO, which is composed of representatives of the 50 state CIOs. Coupling well with other models, this model has broad applicability throughout both commercial off-the-shelf and open-source enterprise environments. Because it is derived from ongoing state-level CIO efforts, this framework remains a living body of data.

**Tip:** As with governance methodologies, elements from the formal frameworks can be combined in order to better meet the needs of each enterprise’s unique setting. As an example, the State of Michigan makes use of
elements from both the Gartner and NASCIO frameworks in the continuing development of its Strategic Plan, with very positive results to show from this effort.

Many mappings exist between the various enterprise framework models to better allow translation and combination of their useful elements. With attention to potential source bias, they can serve as an invaluable resource when attempting to identify boundaries within an unplanned ad-hoc enterprise seeking focus.

**Project and Program Management**

A common fallacy encountered in many organizations is the idea that technology must drive business and operational mandates. Business goals must always drive the logical decision making that produces a technological outcome, or the technology will become an end unto itself rather than a means to facilitate business operations. Aligning these requirements is the most important task to be managed by chief architects as they develop their vision and then implement it into real-world elements. Beyond documenting requirements and technology solutions, architects must also engineer metrics and measurement criteria into each element to be able to identify and illustrate the benefits or costs incurred during implementation.

This is not, by any means, the end of the requirements for an enterprise architect. The chief architect must also plan each element with clear goals, milestones, and completion criteria before seeking or providing approval for each stage. The architect must be the final arbiter when negotiating solutions to conflicts that arise during planning, acquisition, implementation, and adoption of each change—which relies far more heavily on the project management skills of the architect than on detailed knowledge of the technologies involved. Ultimately, the architect must plan, design, guide, and monitor the ongoing fluid process of development throughout the enterprise and all of its associated business elements and technologies.

While the process of enterprise architecture never ends, individual elements must be clearly identified with a scope, planning guidelines, completion criteria, and other elements common to discrete project management. Leadership, communication, negotiation, problem solving, resolving cultural impact, meeting regulatory mandates and standards, quality control, and other aspects of formal project management play a critical role in the success or failure of a chief architect’s contributions. Operational elements within small and medium-sized enterprises may not always need a formal scoping document, formal change control committees, and a detailed risk analysis for each tiny change that may be enacted. However, knowledge of these techniques is vital to knowing when they have become necessary—particularly as the enterprise scale increases.

In addition to operational technology management, the chief architect must be comfortable in some type of formal project and program management methodology. It is important that any other lead architectural roles also involve these principles, but they are imperative for the central architectural role. Note, however, that there are many different formal project management styles. General project management techniques often bring benefits beyond focused-methodology management techniques, such as those that focus closely on quality control or documentation mechanisms specific to particular technologies. More general techniques often address business drivers that may factor into long-range planning without being obvious in short- and mid-term architectural assessments.

**BEYOND BASICS**

The chief architect must always be looking toward identifying trends and emerging strategies that will play a part in the next cycle of enterprise evolution and the ones beyond. The architect must have an understanding of the impact of decisions made in creating and implementing the central vision, to avoid closing doors not yet even glimpsed ahead. This can at times be a journey fraught with peril, as many unanticipated consequences can arise from small differences in technology.

**Language Standard**

The adoption of a programming standard, such as the Java J2EE language, works very well in the resource-plentiful PC environment but may not always scale well into mainframe OS architectures, where thousands of processes may be sharing a vast but not unlimited resource pool. Even though the language can perform within both environments, the manner in which applications must operate can vary widely in such circumstances. Similarly, the selection of an object-oriented programming language (Java, .NET) can create disruption if developers have previously used a traditional language (ANSI C, COBOL), because of the fundamental differences in how these types of programming languages transfer information.

**Operational Environment**

Environments that must support real-time operations rely on solutions different from those required for high-volume transactional processing, distributed computing, or detailed business intelligence analytical processing. The data structures necessary to support each type of
operation differ in terms of resource consumption, scalability, data throughput, metadata organization, 24/7 availability, security, recoverability, and many other factors. Beyond the data structures, user interface design, reporting, and other tools will vary widely among tasks, and decisions can have a strong impact on many of these tasks. Selection of an operating system platform, user office suite, application suite, data management solution, programming language, development suite, and all of the other elements of the technology implementation plan can amplify or negate business drivers present in each scenario.

Virtualization

The potential for technology virtualization seems almost limitless as enterprises take advantage of huge resource and storage pools, enhanced network bandwidth, and improved management utilities. Users can remotely access their desktop workstations from anywhere in the world, while server farms and their attendant power requirements are collapsing by half or better as hardware server sprawl is being channeled into a smaller number of powerful servers supporting software system emulations.

Virtual storage solutions can make use of every scrap of space throughout an entire enterprise, while others may ensure that necessary data is available on the best connected server by replicating a virtual volume throughout the enterprise. Combinations of high-performance computing strategies and service virtualization have led to the virtualization of entire network infrastructural elements into “the Cloud.” These solutions promise incredible advantages in utilization, efficiency, and cost management—and the decisions made will affect their availability down the road.

Mobile Technologies

As network bandwidth and portable computing power increase, the office is leaving a physical building and becoming more a matter of availability anywhere, anytime. Weather, sports, navigation, and other personal interest data is readily at hand, while network management and server console access no longer requires anything more than a WiFi hotspot at the local coffee house or a cellular data link from the back nine on the local country club. The ability to remotely access and manipulate information that remains safe on its host system can help to alleviate business issues for corporations working across national boundaries that may restrict where information may be exported. The ability to virtually access supercomputing power from a mobile handheld device is one of the most powerful causes for change in the modern enterprise.

Service-Oriented Architecture (SOA)

As remote access and network connectivity improve, many enterprises are adopting the service-oriented approach to application development. By making use of Web service standards such as SOAP and Universal Description, Discover and Integration to pass information between applications, programmers can work simultaneously on different parts of a larger application without having to worry about using the same language or parameter-passing mechanisms. All that is required is that a set of standards is used for the information exchange itself—often a variant of the Extensible Markup Language specification. This has the short-term benefits of both rapid development and platform independence that allows a heterogeneous enterprise to take advantage of solutions developed using this methodology. However, the varied nature of development and the distributed processing potential of SOA solutions can complicate technology modernization and disaster recovery planning.

Whatever’s Next

Emerging developments in optical and quantum computing offer glimpses into new mechanisms for cryptography and data mining. The synthesis of asynchronous e-mail and threaded discussion boards with synchronous chat and instant messaging systems is extending into data-based communications for voice-over-IP and teleconferencing, where a shared whiteboard can be used to share doodles across a dozen participants in a dozen different countries. These issues and more must be considered by the chief architect, so that strategies will already be planned if the new mechanisms later reach commercial viability. Built atop existing decisions, the enterprise architect must make choices that will offer several options later.

SUMMARY

In order to succeed, an enterprise needs a coordinated architectural vision. Ultimately, this requires that someone be responsible for creating the vision, presenting the implementation, and resolving the inevitable conflicts that follow change. The architect must be knowledgeable in a wide range of technologies, IT governance, existing architectural models, project management methods, and a wide range of both “hard” and “soft” skills needed to gain support and engagement in the process of turning a vision into a functional enterprise.

The architect must always be learning, looking to the future, and communicating with all involved parties. Enterprise architecture is not a goal; it is a process for
implementing long-term business requirements within an environment of constant change.

**BIBLIOGRAPHY**

1. The IT Infrastructure Library (ITIL), http://www.itil.co.uk.