



Part I

Foundation Concepts

CHAPTERS

- 1.1 Why EA and Why This Book
- 1.2 Enterprise Architecture Concepts

- 1.3 Enterprise Architecting



I. I

Why EA and Why
This Book

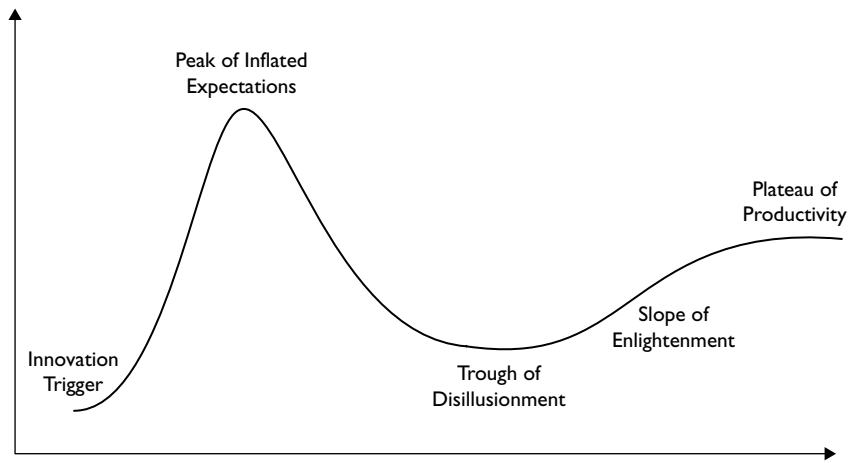
The information technology (IT) company Gartner Inc. (Fenn & Raskino, 2008) has discovered a pattern of hope and disappointment that has plagued almost every highly touted innovation and its adoption by otherwise highly capable and successful enterprises. Gartner coined the term “hype cycle” for this phenomenon. This cycle begins with initial enthusiasm and inflated expectations; results in disappointment when expectations are not met; and ultimately leads to the realization that value can still be realized, albeit more gradually and over a sustained period of time (see Figure 1.1-1). Nowhere was the existence of the hype cycle more evident than in the area of information technology.

Some of the innovations that drove this high sense of inflated expectation and are still fueling the hype cycle are

- Increases in processing speeds and decreases in cost
- Increases in memory size and decreases in cost
- Ubiquitous use of computing through small, fast, cheap processing
- The emergence of the computer as a personal appliance
- Increases in communications speeds and decreases in cost
- Increases in processing capability and improvements in processing algorithms
- Cheap, ubiquitous computer devices
- Technology miniaturization and portability advance

FIGURE 1.1-1

Gartner's
Hype Cycle of
Innovation



- Rapid improvements in absorption of technology among the nontechnical population
- Fusion of business and consumer use of IT

The phenomenon of enterprise spending in this explosion of technology innovation was that of a child in a candy store with a full wallet. Enterprises blithely purchased technology products—sometimes at the individual level, often at the department level, and very rarely at the enterprise level.

Out of the realization that enterprises were spending out of fascination with technology without a clear cost/benefit equation, industry took the first step: appoint a single person and organization to account for and manage centrally the burgeoning IT budget that was being spent in many places within the organization. This function, called the Chief Information Officer (CIO), became the person responsible for approving IT purchases and also the person accountable for the alignment of IT investments to the business of the enterprise. Just as other enterprise costs were measured and expected to be in line with the scale and operations of the enterprise, IT spending was being held to the same standard.

The innovative trigger for the enterprise architecture (EA) hype cycle was the clear emergence that EA was the tool of choice for matching IT investments to the business that they supported. EA allowed the traceability of IT investments to enterprise factors in terms of the activities, locations, organizations, products and services, lines of business, business areas, strategic goals and objectives, and the transformation processes of the enterprise. A seminal work in this area was the methodology for “Enterprise Architecture Planning” by Steven Spewak in the early 1990s, which was quickly adopted by many. Enterprise Architecture Planning (EAP) and other frameworks are discussed in Chapter 1.2.

Emergence of EA

This section presents a few examples of events that have shaped the evolution of enterprise architecture over the years.

Early 1990s: Emergence of the CIO as an Executive Role and a Seat for IT at the Table

During the same early 1990s period, the new role of the Chief Information Officer was also taking shape as enterprises realized the strategic advantage of the information that

they collected in the course of running their businesses. Many industry studies during this period that outlined the top ten challenges for a CIO consistently spoke about two persistent problems: (1) the growing application backlog as the IT shop struggled with rolling out applications fast enough to keep pace with changes in the business and the environment and (2) the need for alignment of IT expenditure to the business. The CIO was in the same executive meetings with the Chief Executive Officer (CEO), the Chief Operating Officer (COO), and the Chief Financial Officer (CFO), and the questions that came from such meetings were not technology-related but rather on how technology created opportunity and how a business case for a technology expense is to be made.

1994: Federal IT Expenditure Was on the GAO's Radar

The need to establish alignment came from the realization that IT had become a significant expense, but IT projects were mushrooming from sheer demand and it was not possible to corral the various categories under which IT money was being spent. In 1994, the Government Accounting Office (GAO), later renamed the Government Accountability Office, surveyed best practices in commercial companies that were able to account for their IT spending and justify the spending by explicitly showing the connection between the IT and the business or alignment. The GAO recommended that the federal government—which was also spending large amounts of money on IT—embark upon measures that would address the issue of IT alignment to the business as well as measures to help architecture solve the problems with stovepipe projects that duplicated effort, created redundant capabilities and data, and required a huge cost in post-implementation interoperability measures such as data exchange interfaces to stitch the output of stovepipe projects together. The GAO also recommended that a formal position of CIO be established within agencies to provide accountability for IT expenses and that all IT expenses be visible to the CIO from a vantage point so that agencies could aggregate the costs of IT and report costs more accurately.

It is important to note that the interest of the GAO, from its very charter and composition, was primarily on the financial aspects and governance of IT, not so much on the burgeoning complexity, the increasing backlog, and the plethora of expensive interfaces that were bandaging systems together haphazardly and delicately.

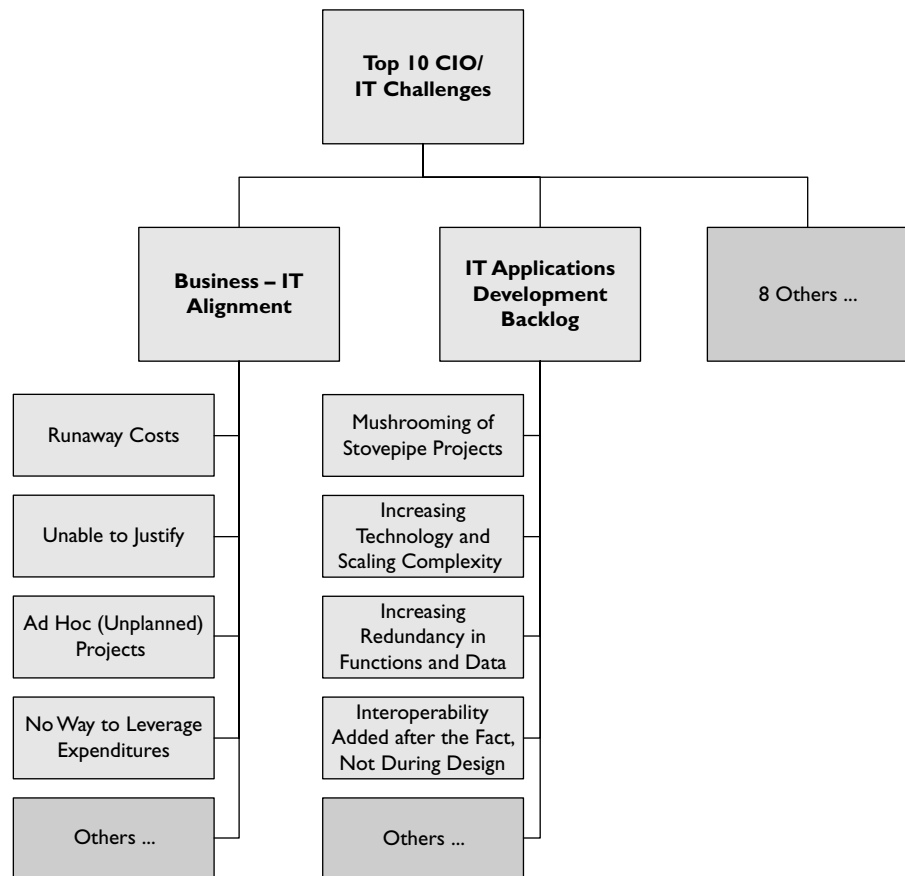
The corresponding movement within the commercial world to govern financial investments was being developed by the International Council on System Engineering (INCOSE). Control Objectives for Information Technology (COBIT)

was one of the best practices and standards that emerged as a result of applying investment and governance measures to IT in a manner similar to other business functions.

1996: Clinger-Cohen Act: The ITMRA Compliance Threat

Congress took to heart the GAO's recommendations and passed the Information Technology Management Reform Act (ITMRA) in 1996 (see Figure 1.1-2). This is also more famously known as the Clinger-Cohen Act (CCA) after its sponsors.

FIGURE 1.1-2
CCA
Recommendations



The CCA mandated the following, under penalties of law:

- Every agency must publish an enterprise architecture that details the business structures of the agency.
- Every agency must appoint a formal position of Chief Information Officer responsible for all IT operations and accountable for all IT expenditures.
- The CIO is responsible for the production of the agency enterprise architecture.
- All agency IT investments must be tied to the enterprise architecture. Business cases must be produced for major investments. No money would be available for initiatives that were not supported by a business case.
- The Office of Management and Budget (OMB) is responsible for governing the provisions of the CCA and maintaining the overall Federal Enterprise Architecture.
- The OMB will monitor and govern IT investments.

What About the Nonfinancial Issues That Plagued the CIO?

The task of dealing with the nonfinancial IT problems that were addressed neither by the GAO recommendations nor the CCA was the subject of much discussion within the Federal CIO Council. An architecture working group (AWG) was formed to explore the use of architecture techniques to ameliorate the issues of interoperability, complexity management, the elimination of redundancies, the expediting of IT to market, and the reduction of the backlog. The framework first articulated by John Zachman and the EAP techniques first articulated by Steven Spewak were the early foundations for the discipline we call enterprise architecture.

1999: Federal Enterprise Architecture Framework (FEAF)

The National Institute of Standards and Technology (NIST) was responsible for developing standards for this new discipline called enterprise architecture by the publishing of the Federal Enterprise Architecture Framework (FEAF) in 1999. The primary basis for the FEAF was the Spewak EAP. Spewak's technique was based on defining an as-is view of the current IT infrastructure and postulating a vision of the future or "to-be" state. He also recommended that the representation of the as-is

view be performed by recording the current inventory of systems and the current structure of business operations. The representation of the as-is view is analyzed to distill the enterprise architecture which is a combination of the data architecture, the applications architecture, and the technology architectures respectively.

1999–2000: The Y2K Threat

At the end of 1999, enterprises were mortally afraid of the change in millennium. They feared that the data they had collected over the decades would become unusable because the year was stored in two digits (with an implied prefix of “19” for the 20th century) and that the calculations that computed differences and added years to a baseline would need to factor in the leading two digits as well. There was a flurry of “remediation” activity where enterprises compiled a list of all applications, short-listed the ones that contained two-digit date fields for remediation, and then established and executed plans to change the code and test the applications with haste.

At the end of 1999 and into the year 2000, few cases appeared, if any, of catastrophic consequences from a bad choice in date storage format. But the benefit of the inventorying exercise is that enterprises now had authoritative lists of IT applications that they could point to, price out, and establish transformation and modernization roadmaps for.

2001: The Disaster Recovery Threat

When the terrorist-flown planes hit the two World Trade Center towers, many enterprises that were doing business at that location were affected. And destroyed along with the towers were computer records, processing equipment, people, networks, and facilities. Recovery in many cases was arduous and, in one case, disastrous. The firm of Cantor Fitzgerald, located in the North Tower, never recovered from the devastation of its facilities. Continuity of operations (COOP) planning is an important enterprise process that requires a holistic view of systems, people, facilities, information, and other items—the architecture elements that are typically part of an enterprise architecture representation. And the disaster of 9/11 brought home the importance of baselining the enterprise architecture—to resume or transfer operations smoothly in the event of disaster.

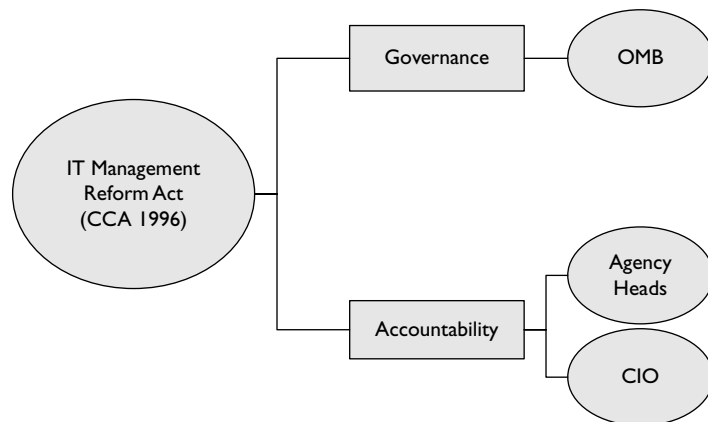
Today: Current Enterprise Challenges

In today's environment, as enterprises become more and more dependent on information to fuel their enterprises and information technology to propel their enterprises, large and small, they face a number of challenges that affect their planning environment; their ability to make decisions; their ability to assess, evaluate, and prioritize the many alternatives; and their ability to face new pressures from regulation, globalization, intense competition, and dramatic changes in their environments. Many enterprises have not been able to address these challenges successfully, as the evidence of a long list of failed multimillion-dollar IT initiatives seems to convey. The complexity and scale of some of these projects in the domain of information technology, though comparable to large civil engineering works or the building of airframes, do not appear to have benefited from a holistic, structured planning approach combined with an integrated, well-managed implementation approach that leverages planned risks to choose good design trade-offs. In short, the use of architecture for the planning phases is required to deal with the complexity and scale of the implementation phase. A systematic application of architecture is therefore the subject of this textbook.

Figure 1.1-3 shows how the Clinger-Cohen Act forces the responsibility of developing an enterprise architecture down to each federal agency, holding the Agency head responsible for the submission and instructing the Chief Information Officer to develop the enterprise architecture for each agency while holding the Office of Management and Budget accountable for the governance of the capital spending based on the enterprise architecture for the entire federal enterprise.

FIGURE 1.1-3

One of the Top Ten Challenges for the CIO: Alignment of IT and Business



The following are a few of the many challenges that enterprises face today:

- Unacceptable numbers of failed large IT projects.
- The need for a faster tempo of operations, resulting in the need to detect and eliminate latencies and reduce the length of the connection to the customer. In the cases of military operations, the need to reduce latency of intelligence, imagery, and other situation assessment and operating picture elements to provide better operational planning and execution capabilities.
- Mergers and divestitures where new parts of the IT environments are acquired from merged enterprises, bringing in diversity in standards, information representations, architectures, cultures, and equipment.
- Globalization where parts of the workforce are located in different regions of the world, with their own cultures and local constraints on power, equipment, languages, and other factors.
- A changing workforce with new ways of interaction and collaboration; different expectations of their work environment; and increased exposure to computers, applications, information, and networking.
- Increasing expenses in information technology as a whole despite dramatic decreases in per-unit cost of processing and memory because of labor, the need to control obsolescence, the evolution of standards, and the need to provide constantly faster and faster processing and increasing memory.
- Increases in the complexity and scale of IT projects both in terms of scope and reach. Consequent needs for orchestrating work outputs of large numbers of professionals located all over the globe to complete projects successfully. Inability to fully assess the impact of complexity and scale at the planning or implementation level.
- Rampant adoption of technology based on competitors' use of technology, vendor push, as well as the pressure to align with contemporary standards.
- Rampant outsourcing, based on financial analyses, competitive price pressures, as well as the current enterprise focus on the core mission and the decision to outsource all noncritical supporting services.
- Dramatic opportunities posed by evolving architecture styles such as service-oriented architectures and cloud computing that change the way systems have been built in the past and introduce new planning challenges of migrating ponderous current systems infrastructures to a different type of architectural future.

The Case for Enterprise Architecture

The famous Finnish architect Eero Saarinen recommends, as a design principle: “Always design a thing by considering it in its next larger context—a chair in a room, a room in a house, a house in an environment, an environment in a city plan”.

A business process—a collection of activities directed toward achieving a business objective—does not stand simply as a list of tasks. It also involves outputs of those tasks, some of which are inputs to tasks belonging to other business processes. The tasks involve performers who are also performers of tasks for other business processes. The performance of tasks requires the existence of organizational relationships that are part of the organization chart for the enterprise.

Although we will formally define and elaborate on enterprises and enterprise architecture, we define *enterprise* loosely as a shared endeavor at any level directed toward a common objective. This definition allows us to cover any level of abstraction—from the executive branch of the U.S. government as one of the three branches of government to the Department of Homeland Security as a cabinet-level organization to the Customs and Immigration Services Agency (CIS) to a regional office located in Philadelphia, Pennsylvania, to a software development effort for a \$41 million software system. An *architecture*, loosely defined, is the structure of components and their relationships. When architecture is applied to enterprise, we get *enterprise architecture*: the structure of components of an enterprise and their relationships.

Communication

Enterprise architecture deals with a *uniform* way to represent the larger context and to *communicate* that context to everyone in the enterprise who is an interested party in some enterprise process. That larger context may embrace elements inside and outside the enterprise and frequently does so.

Notice the emphasis on *uniformity* and *communication*.

In the day-to-day world, communication is accomplished by spoken and written language. There is an assumption that the speaker and the listener are able to understand the same vocabulary and the same sentence structures and grammar, and are able to comprehend the meaning (semantics) of the content of the conversation. Enterprise architecture establishes the common vocabulary for an enterprise, the grammar and the semantics for various stakeholders to communicate understanding of architecture elements, and the relationships among architecture

elements. As we will see in other chapters of the book, this entails the following topics.

Unifying Architecture Frameworks

An architecture framework establishes a set of terms that describe the various types of architecture elements. For example, the Department of Defense (DoD) Architecture Framework defines architecture elements such as Activity, Performer, Location, and Information. It further specifies Activity as performed by humans or as performed by an automated system; Performer as a human, organizational, or notional performer or as a system, a business service, or a software service. The adoption of a single common framework throughout an enterprise ensures that all architecture-related terms are couched using the same underlying terminology. If two collaborating enterprises were to adopt different frameworks, they run the risk of speaking past each other and may need translation to understand each other.

A Rich Set of Models

Once a set of common vocabulary terms is defined, the next challenge is that of developing common grammars or sentence structures that can be commonly understood by diverse groups of people. In architecture work, as in any other area, a picture is worth a thousand words. But a picture is simply a cartoon without a blueprinting discipline that standardizes what the elements in the picture represent. The picture must make consistent use of the same symbols for items of the same type. Similarly, the picture must describe what the connections among symbols mean. Randomly drawn pictures are therefore not the answer to providing a uniform and consistent way to communicate. Over time, many methodologies have evolved that have standardized the blueprinting of models that support the various activities of the enterprise. In Chapter 5.1 of this book, we will discuss some of these methodologies. But for the concern of this chapter, enterprise architecture involves representing architecture elements and their relationships using commonly understood and consistently applied models. This consistency can be achieved more easily by using a common set of automated or nonautomated template-based tools throughout the enterprise rather than relying on individual modelers to conform to common standards.

It is limiting to think of models as simply the artifacts of IT specification, analysis, design, and construction when the scope of modeling is not restricted to software development, but rather the enterprise as a whole.

True, the evolution of modeling technologies such as the Unified Modeling Language (UML) and Structured Analysis and Design Technique (SADT) are directly derived from the software development domain. IT personnel first had to wrestle with the problem of modeling business processes, states, and electronic information to be able to represent the requirements that they were asked to implement within an automated system. In fact, the computer-aided manufacturing program that produced the information definition (IDEF) series of modeling techniques directly piggybacked and evolved the techniques of progressive refinement and decomposition that are part of structured analysis.

The use of models in EA is not restricted to the ones that are used for software or business process modeling. For example, because a stakeholder has a financial interest in a software development as an investment, we need to view the models that are submitted for capital planning and investment control decision making also as legitimate EA models, in the manner we consider data models, functional decompositions, or object class models as representative of the types of models required for enterprise architecture representation. In the U.S. government, in fact, a federal agency is required to submit business case justification in financial terms to the Office of Management and Budget in support of a request to pursue an initiative. Commercial enterprises also frequently have investment review boards that review proposals to start or continue new projects.

Models convey knowledge of a certain aspect of the enterprise to a specific audience who can use them for analysis and to make decisions.

For example, in the U.S. federal government, submission of the A-300 business case justification models to a Capital Planning and Investment Council allows the council to make a decision on which investments to approve and which investments to curtail because of budget constraints or perceived irrelevance to the mission of the enterprise.

Another example, in the U.S. Department of Defense, an information support plan might be developed that describes the burden that a proposed system places on shared infrastructure such as network bandwidth or transactions on a shared database server. Infrastructure planners might use such a plan either to curtail the initiative or to develop the common infrastructure to incorporate the needs of the initiatives.

Transformation

Enterprises are constantly transforming—sometimes proactively as part of a planned improvement or reactively struggling and kicking as a response to changes in environment or competition. In either case, any successful transformation is predicated on the following:

- A clear understanding of the *current state*—what we will call the baseline or as-is state. The as-is state of the enterprise may be represented in many dimensions: current motivation, current set of products and services, current business processes, current organizational structures, current event responses, current operating locations, and so on.
- A clear understanding of the *objective* or *target state*—what we will call the to-be or target state. The to-be state of the enterprise must also be represented in the same dimensions as the as-is state. From this representation comes an understanding of what remains the same, what cannot change easily and must stay, and what must change.
- A clear *transformation plan* or *roadmap* that will transform the enterprise from the as-is state to the to-be state. This plan has to have a timeline that is acceptable within the window of change; otherwise, we risk a transformation that is too late—usually calling for enterprise triage measures. The roadmap generally consists of a number of concurrent initiatives or projects that must all support one another and join to provide the net transformation effect that the enterprise desires.

The representation of the as-is state of the enterprise, the to-be state of the enterprise, and the planning of the transformation roadmap (family of solutions) are all part of the enterprise architecting effort.

Addressing a Family of Problems and Solutions Simultaneously

Enterprise architecture provides a common context for a large family of problems and solutions. Today, the disciplines of systems engineering, software development, and project management amongst others have evolved into detailed bodies of knowledge and practice that have become powerful tools for solving individual problems. But enterprise architecture is the discipline that provides a context for solving, not the individual problem, but a family of problems without the risk of solving one problem only to aggravate or cause another because of ignorance of the overall context in which the problem is being solved.

Systems Engineering

The area of systems engineering has progressed tremendously and developed several successful methods of planning, designing, implementing, testing, simulating, and deploying complex systems. Formal methods are available for capturing

systems requirements, analyzing systems requirements, and building various types of analytical models for representing various dimensions of a system. Systems engineering also deals with design constraints and trade-offs involved in designing for competing constraints. For systems that are built in isolation of each other, these techniques have been more than adequate. The techniques of systems engineering can span the abstraction of analysis to the detail of design and implementation. As the scale of systems has increased as well as the number of design variables, ill-understood coupling of system variables has resulted in unintended design consequences. Systems engineering practice is trying to evolve to meet the need of complex, large-scale systems.

Software Development

The software lifecycle has also been well studied and documented. Most enterprises with a mature software development culture have documented software development lifecycles (SDLCs) with standards, methods, tools, and workflow as well as gates or review points for each phase of the lifecycle to manage cost, schedule, and quality for a software development project. The SDLC that started as a progression of water-tight development phases that delivered results only after a couple of years to a problem that disappeared a year ago has yielded to rapid application development, spiral methods for delivery, and agile development and Scrum techniques—always trying to shorten the gap between the specification of requirements and the delivery of working software.

Project Management

The discipline of project management has also evolved to its current level of rigor and quality with the establishment of commonly understood project management techniques; risk management processes; project reporting techniques; and techniques for managing cost, schedule, and quality. With the emergence of earned value management (EVM) as a measure to assess the health of projects, enterprises are looking at effective project management as a silver bullet. Unfortunately, the root cause of project failure is seldom the failure of project management—it goes deeper to the limited understanding of the complexity of the problem, or the lack of depth during the planning, scoping phase, and many times to the law of unintended consequences.

Arguably, at the head of all of these—which is the point at which an enterprise plans a specific systems development, software development, or project—there are few alternatives for analyzing the portfolio of systems, software development

projects, or initiative projects. This is where enterprise architecture comes in. Enterprise architecture provides a representation means for the planner to analyze a family of projects or initiatives, a family of systems, or a family of services. This analysis is very important because each of the ingredients of such portfolios makes a demand on capital and working capital funds, development resources, management attention, and has different, often disruptive effects on the enterprise while it is under way.

Leveraging Patterns

Patterns are repeatable configurations of elements. In enterprise architecture work, patterns are complex combinations of architecture elements that provide a repeatable design. An example of an enterprise architecture pattern is a fast food franchise that makes burgers. The architectural pattern for the process of making the burger is the combination of the activities involved in making a burger, the skill sets of the personnel involved, training on the operations involved, the specific equipment that is involved in the making of burgers, and any other items required to specify completely all the factors that go into repeatedly making burgers that are consistently the same. Along with the specification of the “hard items” is also a recipe for temperatures, times, ingredients, and preparation instructions. The success of the franchise in turning out repeatedly consistent product is predicated on standardizing all of these elements and not allowing variation. This is frequently a condition that the franchisor places on a franchisee.

Enterprises can leverage patterns for expanding their operations in a reliable and repeatable manner. At the same time, they can exploit regional and local differences to customize patterns for better fit in areas where such diversity is important.

For example, developing an enterprise architecture enables an enterprise to identify and use patterns in supporting standardization efforts, improving business processes, or scaling up operations.

Decision Support

All enterprise operations and transformations involve decision making. Decision making involves looking at information. Decision making also involves looking at candidate alternatives and determining a course of action based on selecting the best of these alternatives. Decisions are often made with limited information and the assumption of calculated risks. Decision making at the planning level often can have a significantly larger impact on cost, schedule, and quality than decisions made lower down in the implementation phases.

Enterprise architecture is a knowledge base for the executive decision maker at the planning level. By providing the right information along with the assessment of impacts, the enterprise architecture can provide a basis for informed decision making that increases the probability of plans that will successfully turn into implementation reality.

As executives and decision makers start using the enterprise architecture information in their decision making, they will find (a) needs for additional information that the enterprise architecture representation has not captured, (b) new uses of enterprise architecture information, (c) shortfalls of critical information that is required but is not available, or (d) inaccurate and out-of-date information that is not fit for use in the planning without introducing new risks from inaccurate data.

As we will see in Chapter 2.2, scoping and planning the purpose of an EA effort are critical to managing the success of the EA efforts and the usefulness of the information delivered to decision makers.

Enterprise Architecture Certification

The haphazard and often unpredictable rate of progress of enterprise architecture thus far has been directly attributable to individual efforts and specific efforts rather than a more general widespread awareness and competency base that can produce repeatable and workman-like results.

Development as a Practitioner-Driven Discipline

We presented the hype cycle in the introductory section of this chapter. EA has had its own hype cycle. The triggering event was the passing of the Clinger-Cohen Act and the consequent gold rush to the enterprise architecture field to satisfy the sudden demand for enterprise architects.

Enterprise architecture developed as a practitioner-based discipline rather than being grounded in any specific academic discipline. It began with the need for an integrated IT architecture that encompasses multiple views of the enterprise to reduce redundancy and gaps in investments and assist IT strategic planning. It soon became apparent that to accomplish this goal, EA required a holistic enterprise view that not only includes business perspectives and requirements but needs to be driven by them. EA emerged as a way for an enterprise to be fully integrated and in communication with itself and its business partners to enable all enterprise stakeholders both to manage complexity and to understand and handle the impact of any proposed change in business strategies, information, and technology at every level.

This shift from an IT to an enterprise-wide program raises serious issues regarding what needs to be incorporated in an EA education and what skills and knowledge must be tested to ensure that someone is a fully trained and proficient enterprise architect. The education of an enterprise architect involves being competent in multiple disciplines. An architect must be a systems thinker with an understanding of organizational development, business processes, strategic planning, governance, program management, and related topics relevant to the views of the business planner and owner. At the same time, the architect must have expertise with the wide range of methods relevant to building information, data, application, and technical models.

The Need for Multidisciplinary Skills

The education of an enterprise architect involves being competent in multiple disciplines. These comprise, on the one hand, the contributions to a wide range of fields contributing to the study of organizational theory. These include social and business sciences, such as anthropology, business administration, communications, linguistics, public administration, and political science. These disciplines introduce to EA such areas of involvement as the study of values, ethics, complexity science, organizational behavior, organizational structure, organizational development, communication and social networks, communities and networks of practice, leadership, management theory, organizational culture and climate, industrial psychology, policy analysis, semantics and structures of meaning, and belief systems.

Relationships to Other Bodies of Knowledge

Concomitantly, the discipline of EA is connected to general systems theory and systems engineering. This introduces systems engineering, engineering management, technology management, computer sciences, information sciences, and operations research as input fields. These contribute to process analysis, technology assessment, requirements analysis, cost engineering, quality control, project management, environment scanning, business reengineering, enterprise engineering, and Six Sigma.

With inputs from the organizational and systems sciences, a number of emerging fields have developed within EA that are relevant to understanding the influence it has on enterprises of every size. These include information resources management, information assurance, security management, and records and data management, enterprise resource planning (ERP), service-oriented architecture (SOA), and cloud computing. In turn, EA contributes to a range of new, emergent concepts

including solutions lifecycle development, portfolio management, capital planning and investment control, knowledge management, e-government and e-commerce, business transformation, ubiquitous computing, as well as DoD-relevant concepts such as the global information grid and the Joint Capabilities Integration and Development System (JCIDS) (discussed later in this book).

As we shall see, EA has both a hard side and a soft side. As Marc Lankhorst noted next to its architecture, “which could be viewed as the ‘hard part of the company,’ the soft part, its culture, is formed by its people and leadership and is of equal if not higher importance in achieving (organization) goals” (Lankhorst et al., 2009: 9). Yet organizational behavior, culture, and other aspects of the soft side of a company can be modeled and incorporated into EA. This will be discussed later. The point now is that EA draws from a wide body of knowledge of which an enterprise architect should be aware.

EA Knowledge and Expertise

By examining government and industry initiatives and the growing appearance of EA within university departments, we can address what constitutes a viable EA body of knowledge and the particular skills in systems thinking, program management, enterprise modeling, and organizational development that are involved.

In this context, we must consider how these competencies can be realistically tested to certify architects to ensure that they possess both knowledge about EA and knowledge of how to create enterprise architecture. This is one of the major goals of this text.

In a recent report the Gartner Group warned of a danger that due to the recent proliferation of training and certification programs, the viability of the certification process could be questioned. A related problem is in how enterprise architecture has been treated in academia. Over the past few years, EA education has found its way into various curricula but as a part of a degree program rather than as a stand-alone degree program. EA is now moving from being taught as a module in a particular course to being a course on its own to being several courses in a program allowing students to major or minor in it. This advance, however, contains a risk of EA being subsumed under a single technical IT discipline, thus leading to a lack of acknowledgment of the range of competencies recognized by practicing EA practitioners. This portends a possible retreat from the advances that have been made in moving EA from being IT-centric to a true enterprise-level concern.

In this chapter, we consider the knowledge, skills, and competencies required of architects and what type of processes are needed to test their acquisition for

certification effectively. The recognition of the broad body of knowledge that this entails is behind several current initiatives to define competency framework for certification of architects by such bodies as the DoD Office of the CIO, the Academic Coalition of the Industry Advisory Council, the Open Group, and Center for Advancement of Enterprise Architecture as a Profession (CAEAP). After examining these government and industry initiatives and the discussions within university departments as EA programs are being developed, we then address what constitutes a viable EA body of knowledge and the particular skills in systems thinking, program management, enterprise modeling, organizational theory, and organizational development that are involved. In this context, we consider how these competencies can be realistically tested to certify architects to ensure that they possess both knowledge about and knowledge of how to create enterprise architecture.

What It Means to Be Certified

Certification in most areas refers to a formal procedure to recognize the knowledge, skills, and experience of members of a profession. This is most often done in part by asking candidates to demonstrate their knowledge through some form of examination procedure. In most but not all instances, certification is done through some external body or organization that is recognized as a credible body to assess capabilities. In many disciplines, such as medicine, education, and law, this certification is under the authority of major professional organizations and/or governmental entities. There are also a number of technical certifications that are promoted by industrial organizations that test for competencies with a defined body of knowledge and skill sets. These range from certifications in specific products such as those offered by Microsoft, CISCO, and Oracle to the mastery of a set of approaches that are often incorporated into the curricula of various university departments. These include such fields as program management, Six Sigma, information assurance, and the like.

As the discipline of EA emerged in the late 1980s and became mandated by governments—such as when the Clinger-Cohen Act of 1996 required all United States federal agencies and the Department of Defense to use architectural principles to manage their information technology resources—a critical need emerged to ensure that those responsible for doing EA had mastery of the body of knowledge and skills involved. It was with this challenge in mind that in 2001 Felix Rausch and Beryl Bellman founded the FEAC Institute as the first organization to offer formal EA education and certification in the Federal Enterprise Architecture (FEA) and the Department of Defense Architecture Framework (DoDAF). FEAC originally

was an acronym for Federal Enterprise Architecture Certification, and later expanded its definition to Federated Enterprise Architecture Certification, to include, in addition to EA in the federal sector, the knowledge, lessons learned, models, and capabilities of other types of enterprises.

The need for education and certification in EA is exemplified in a true, albeit humorous, story about one of our FEAC students early in the Department of Defense Architecture Framework (DoDAF) program. The major FEAC programs in FEA and DoDAF both entail 200 contact hours of mixed mode or blended education combining face-to-face with online learning given within five graduate-level courses affiliated with FEAC's university partners. Students in the program receive continuing education units or, upon payment of tuition fees, receive from FEAC's partners graduate engineering units that are applicable to a number of graduate programs. The program is delivered by having students attend the first 40 hours in a weeklong face-to-face instructional setting. The students then participate for three to four weeks in the FEAC Virtual University, where they take part in online discussions, submit individual and team-based assignments, and take examinations on the topics involved in the first part of the program. Students then return for a second face-to-face, 40-hour, weeklong session, after which they once again take part in the Virtual University. During the course of the entire program, students undertake an actual enterprise architecture project where they build a set of relevant EA models (artifacts or products) that comprise part of a real EA project or practicum.¹ At the end of the program, students take a formal written certification exam (composed of essay and diagrammatic answers) and then formally present and defend their practicum before a faculty panel and their fellow students.² They also write up their practicum projects as theses or narrative reports that are targeted to their internal or external EA customer.

In the case of the previously mentioned DoDAF example, the student told several of our faculty at the end of the first face-to-face session that he was especially concerned that at that point he had little knowledge of EA but was assigned to a major initiative in his branch of the military. The faculty assured him we would work closely and mentor him throughout the process to ensure the quality of his practicum and that he obtained the requisite knowledge and skills. When the student returned for the second face-to-face session a month later, he expressed his excitement and gratitude for the attention the faculty provided him, but said he

¹ These models are presented in Chapter 1.3.

² Sample tests are included on the CD.

was now especially concerned because during the intervening period, he had been appointed Chief Architect and now had to do really well.

This is more than an entertaining anecdote, as it seriously reflects what many architects experienced particularly in the earlier part of the decade. EA was not a subject taught within an academic discipline and was at most a topic mentioned only in the context of some other course. Architects had to learn on the job and quickly master the frameworks, views, models, artifacts, and deliverables that were specified in a number of mostly government-published documents. The FEAC Institute was founded upon and is organized to provide this education and certification. FEAC offers both education and testing of EA competencies and is the only certification that requires individuals to build actual enterprise architectures as part of the process.

Other EA certifications are on the market, and former FEAC faculty members established several of them. Although these programs offer training in EA, they do not require students to demonstrate how to actually create EA models and artifacts and be evaluated in their competencies for doing so. These programs attest at different levels that the student has attended training and can discuss what is involved in developing enterprise architecture. What distinguishes FEAC is ability to both “talk the talk” as well as “walk the walk.”

In this book, we prepare readers to meet the requirements for FEAC certification. As a textbook for FEAC courses, the book includes content for the five FEAC courses in its programs. These range from EA framework basics, planning for architecture development and use, EA implementation, and EA integration. In this context, we present the essential components and uses of enterprise architecture and describe the large set of models or products that are used to build them. These include the views and all of the models and products comprising the current Department of Defense Architecture Framework (DoDAF 2.0), which we have found relevant to every architecture framework. We also discuss a range of reference models relevant to enterprise architecture, beginning with a focus on those mandated by the United States Office of Management and Budget, and then relate them to reference models being developed for the DoDAF as well as those used in other frameworks such as The Open Group Architecture Framework (TOGAF).

We also discuss how to engage in EA analysis. This includes the types of queries and reports that can be generated from enterprise architecture, especially facilitated by the use of one of the viable EA tools. We also discuss how to integrate EA with a number of other critical processes, such as portfolio management, capital planning and investment control (CPIC), information assurance and security, and the relevance and use of Federal Segment Architecture Methodology (FSAM), as well

as Rapid Segment Architecture Methodology (RSAM), developed by Health and Human Services, which tailors FSAM for shorter timeframes and projects. Although several of these approaches have been developed for the United States federal agency context, they are also relevant to both other governmental (state and local as well as other international initiatives) and commercial contexts.

The Relevance of This Text to Other EA Certifications

In 1984, John Zachman, while at IBM, developed the first IT architecture framework, which he continued to develop into the most recognized enterprise architecture framework, the Zachman Framework (Zachman, 1987). The Zachman Framework provides five different views (planner, owner, designer, builder, and subcontractor), each addressing the set of Aristotelian interrogatives (what, who, when, how, where, and why) to create a set of 36 cells or primitives that contain different possible types of models. The set of possible models within a cell constitutes composites that are integrated sets of products for each view. As discussed in more detail shortly, the Zachman Framework is a schema or what he also analogizes to a periodic table of elements for an enterprise. It is not a methodology, but rather a guide for locating work done in various architectures, and serves as a benchmark for comparative frameworks.

For a number of years, Zachman resisted various attempts by various organizations to purchase his framework as their intellectual property. When one of the authors of this book first met Zachman while the former was a contract consultant to NCR, the company wanted to negotiate such a relationship. Zachman instead maintained the integrity of his concept, wanting it to serve as a general basis for all architectural work. NCR developed its own version, the Global IT Planning Methodology (GITP), which, like all other frameworks, is comparable to the Zachman Framework. In this sense, the Zachman Framework is a kind of meta-framework.

Because the Zachman Framework is ubiquitous, it did not require or specify a particular body of knowledge relevant for a certification process. The importance of Zachman's work cannot be understated, as it is foundational in the development and evolution of EA. For this reason, it is discussed in some detail in this book with an orientation toward how other frameworks can be mapped to it and enable cross-framework comparisons.

Zachman Institute Certification

Although virtually all EA certifications recognize the Zachman Framework, in 2005 Zachman, in collaboration with some colleagues, developed a certification program that entails attendance at a weeklong workshop and also added a classification schema based on years of experience and recommendations by other architects to achieve different levels of certification status. Again, this text covers the essential concepts relevant to those workshops and addresses the body of knowledge that Zachman and his associates consider relevant to being Zachman-certified.

The Open Group Certification

During the 1990s, The Open Group (TOG) began developing The Open Group Architecture Framework (TOGAF). This led to a number of versions of their architecture beginning with a proof of need in 1994. The “proof of concept” for the framework in 1995 resulted in the first version. This was followed by continued developments of the parts of TOGAF to be discussed later in this book, which included the Architecture Development Methodology (ADM), the Architecture and Solution Continuum, Reference Models (RMs), and so on. This resulted in the first major rollout in 2001 of TOGAF 7, which was based on the DoD C4ISR (command, control, communications, computers, intelligence, surveillance and reconnaissance) and the DoD models from the Command and Control System Target Architecture (C2STA).

In 2001, the same year FEAC was founded and established its unique and rigorous certification programs, TOG initiated its certification training courses for version 7. Students obtained certification by attending a weeklong workshop. As TOGAF developed over the next several years with TOGAF 8 and TOGAF 8.1, the certification process was roughly the same. Then, in February 2009, TOG introduced TOGAF 9 and for the first time required that students pass two levels of certification examinations, administered by Prometrics, to achieve certification. The exams are multiple-choice questions. Again, this text provides an overview to help readers who want to take the level 1 and level 2 exams.

Certification Process and Certificates

An important distinction needs to be made at this point between certification and certificates. Many of the “certification programs” on the market are in reality “certificate programs,” where one is given a document attesting to the completion of

a workshop or even some course of training. By “certification,” we mean in contrast that after students undertake the requisite training and education in a field, they are examined according to their competencies and ability within it. This is what distinguishes FEAC certification from other programs currently on the market.

FEAC Certification Process

FEAC certifies and ensures that students have demonstrated the knowledge, skills, and capabilities of an architect by creating an actual EA in a practicum thesis project for their work program. This includes creating the full set of artifacts (models) comprising enterprise architecture—with a focus on all of the DoDAF 2.0’s 52 models and views. As discussed later, FEAC uses the DoDAF 2.0 as a comprehensive set of views and models that are also found in whole or in part in the other major architecture frameworks. Besides ensuring that students know how to build effectively the range of models or artifacts involved, FEAC attests that students are able to select the appropriate views and models for an EA that satisfactorily fit their project’s purpose. In this way, FEAC ensures that its graduates both can “talk the EA talk” as well as “walk the EA walk.”

Learning Objectives for This Book

It is the goal of this book to prepare the reader for FEAC certification by covering the major EA concepts and helping the reader to demonstrate competencies in the following areas:

- Be able to select and build appropriate artifacts and models for fit-for-purpose architectures
- Be able to develop EA principles
- Understand and be able to use Reference Models, including the FEA RM
- Be able to create segment architectures using FSAM and RSAM
- Be able to employ TOGAF ADM, Spewak, and DoDAF Six Step methodologies
- Be able to engage in EA analysis and build actual customer deliverables
- Be able to utilize EA for Capital Planning and Investment Control (CPIC) in FEA, Joint Capabilities Integration and Development System (JCIDS) in DoDAF, and Portfolio Management

- Integrate Enterprise Security Architecture into EA
- Be able to utilize EA for SOA and cloud computing
- Be able to conduct EA analysis and queries
- Use EA for organizational transformation

Scope of the FEAC Certification Syllabus

The FEAC certification process is based on the following foundation.

Formal Coursework

Five core courses are taken over a period of about three months that cover the material presented in this book:

- Framework Basics
- Core Models of a Integrated Architecture
- EA Planning
- EA Implementation
- EA Integration

Additional special topics cover contemporary subjects allied to enterprise architecture, such as the DoD JCIDS, Portfolio Management, and SOA.

Formally graded assignments and examinations in each of the individual subjects test the candidate's knowledge of and familiarity with the subject.

Practicum Project

To obtain FEAC certification, students are required to create a portion of an enterprise architecture, which can be part of a larger EA effort for a governmental agency, military organization, or company. Because FEAC works in conjunction with our accredited university partners, this work is done within the timeframe of an academic term of three months. Consequently, it is beyond the scope of our programs for any student to create most organizational architectures. What is important to FEAC is that students are able to demonstrate their understanding of EA and be able to construct real EA models that can be used as actual deliverables

in a larger EA program. However, in some instances, a full EA is possible; these are smaller projects that can be managed within the academic term.

We care that students be able to define clearly an architectural problem that can be addressed using enterprise architecture. Students must be able to select a set of models from those available and with that set build architectural building blocks and artifacts that have real purpose. Consequently, time and project management are critical to the process. Some examples of delimited practicum that our students have developed include the following:

- Global Disaster Response Planning (GDRP) enterprise architecture as the DoD's approach to construct timely and agile crisis action plans that fulfill national security objectives. This practicum modeled the as-is GDRP enterprise architecture effort, including descriptions of the architecture artifacts.
- A Tornado Notification Enterprise Architecture for the National Weather Service (NWS). The practicum provided models detailing the NWS tornado notification process.
- Reference architecture relevant to identity management implementations within the federal enterprise such that it can be continuously aligned and integrated to realize both the citizen-centric performance objectives of the federal government and the simplification of the Federal Enterprise Architecture.
- Communications and data restoration architecture that integrates into and complements the present National Aeronautics and Space Administration (NASA) Enterprise Architecture. The practicum defines a complementary architecture that facilitates the use of NASA Marshall Space Flight Center communication and data restoration services at any NASA facility during a disaster (natural or otherwise).

Questions

1. Discuss what motivates (or hinders) EA development in your organization. What are the key drivers for EA in your enterprise? What are the key demotivators?
2. How does the Gartner Group's hype cycle relate to your organization's adoption of EA? Where is your enterprise in the continuum of expectations

related to EA? What has been the history of your organization with respect to other new technologies, such as SOA, cloud computing, and semantic web technology?

3. What were the early reasons for the emergence of EA? Are those reasons still valid? Are there new reasons driving the need for EA? What are they?
4. What are the provisions of the Clinger-Cohen Act, in brief? What was the driver for the Clinger-Cohen Act? Are the factors that led to the Clinger-Cohen Act pre-1996 still valid today?
5. What is the relationship between enterprise architecture and enterprise transformation? What are the dimensions of enterprise transformation?
6. What is the relationship between enterprise architecture and systems engineering? How are they different?
7. What is the role of project management as a discipline in enterprise architecture? Think of two roles: projects that are related to the development of the EA itself and projects that represent transformation initiatives or investments.
8. What is the relationship between enterprise architecture and the SDLC? What is the role of EA in defining requirements for software development?
9. What are patterns? How are patterns useful in codifying knowledge representations? How are patterns used to replicate entire enterprise structures? Discuss the applicability of EA patterns in constructing, for example, fast food franchises.
10. Why is EA certification beneficial to enterprises? What are the elements of EA certification? How is EA certification as described in this book different from or similar to other certification processes such as product certifications from, for example, Microsoft or professional certifications such as the Project Management Institute?
11. Should EA be restricted to use in information technology planning or is it applicable to all aspects of the enterprise such as the value-added business processes, inbound logistics, outbound logistics, and sales and marketing, for instance? Discuss why, or why not, and present examples of how EA can be used in the IT realm as well as in the business realm.

Reference List

- Fenn, Jackie, and Raskino, Mark. (2008). *Mastering the hype cycle: how to choose the right innovation at the right time*. Cambridge, Massachusetts: Gartner Inc., Harvard Business Press.
- Lankhorst, Marc. 2009. *Enterprise architecture at work: modeling, communication and analysis*. Springer.
- Zachman, John. 1987. *A framework for information systems architecture*. IBM Publication G321-5298.