Chapter 1

Introduction

Note: This handout is derived from Chapter 1 of TOGAF 9.2. It is intended to be used for TOGAF 9.2 Level 1 training.

TOGAF is a framework — a detailed method and a set of supporting tools — for developing an enterprise architecture. It may be used freely by any organization wishing to develop an enterprise architecture for use within that organization.

TOGAF is developed and maintained by members of The Open Group, working within the Architecture Forum (refer to www.opengroup.org/architecture). The original development of TOGAF Version 1 in 1995 was based on the Technical Architecture Framework for Information Management (TAFIM), developed by the US Department of Defense (DoD). The DoD gave The Open Group explicit permission and encouragement to create TOGAF by building on the TAFIM, which itself was the result of many years of development effort and many millions of dollars of US Government investment.

Starting from this sound foundation, the members of The Open Group Architecture Forum have developed successive versions of TOGAF and published each one on The Open Group public web site.

If you are new to the field of enterprise architecture and/or TOGAF, you are recommended to read the Executive Overview (refer to Section 1.2), where you will find answers to questions such as:

- What is an enterprise?
- Why do I need an enterprise architecture?
- Why do I need TOGAF as a framework for enterprise architecture?

1.1 Structure of the TOGAF Document

The structure of the TOGAF documentation reflects the structure and content of an Architecture Capability within an enterprise, as shown in Figure 1-1.

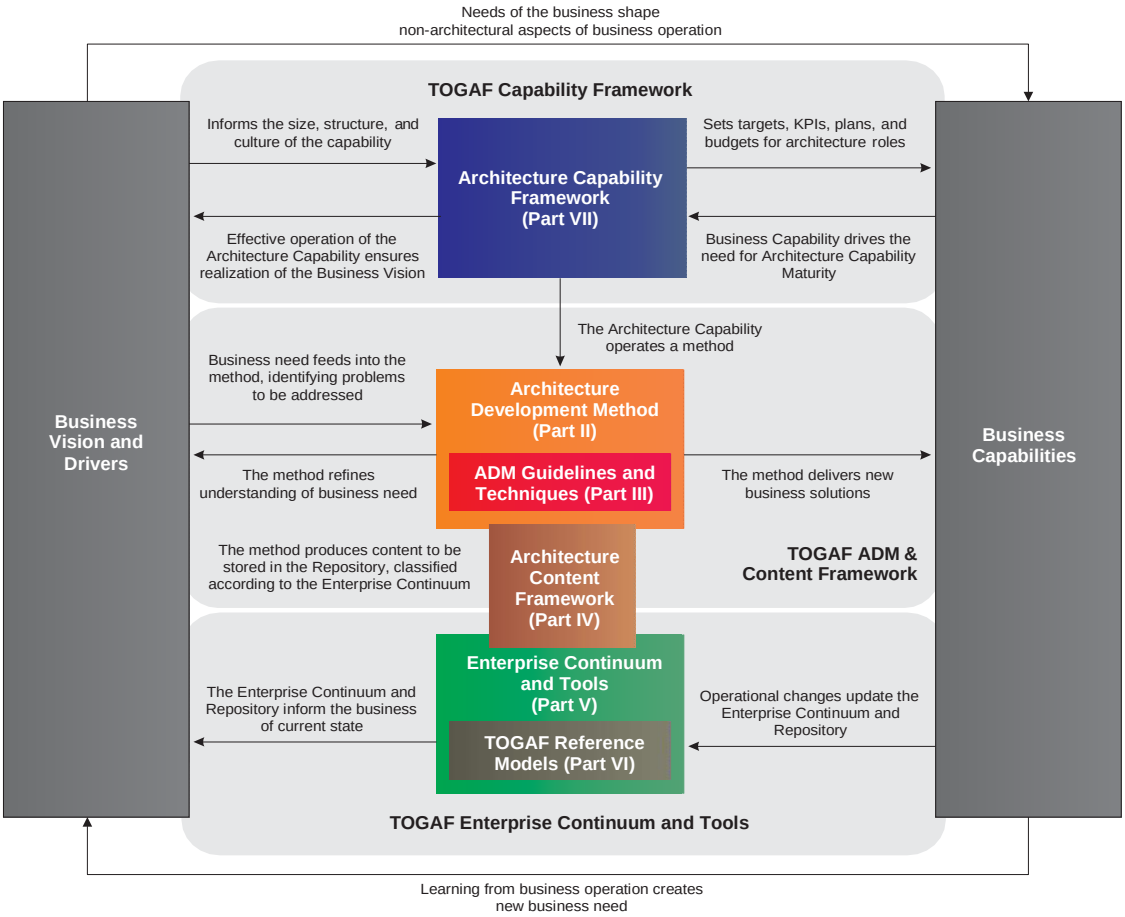

Figure 1-1 Structure of the TOGAF Document

There are seven parts to the TOGAF document:

PART I (Introduction) This part provides a high-level introduction to the key concepts of enterprise architecture and in particular the TOGAF approach. It contains the definitions of terms used throughout TOGAF and release notes detailing the changes between this version and the previous version of TOGAF.

PART II (Architecture Development Method) This part is the core of TOGAF. It describes the TOGAF Architecture Development Method (ADM) — a step-by-step approach to developing an enterprise architecture.

PART III (ADM Guidelines and Techniques) This part contains a collection of guidelines and techniques available for use in applying TOGAF and the TOGAF ADM.

PART IV (Architecture Content Framework) This part describes the TOGAF content framework, including a structured metamodel for architectural artifacts, the use of re-usable architecture building blocks, and an overview of typical architecture deliverables.

PART V (Enterprise Continuum & Tools) This part discusses appropriate taxonomies and tools to categorize and store the outputs of architecture activity within an enterprise.

PART VI (TOGAF Reference Models) This part provides a selection of architectural reference models, which includes the TOGAF Foundation Architecture, and the Integrated Information Infrastructure Reference Model (III-RM).

PART VII (Architecture Capability Framework) This part discusses the organization, processes, skills, roles, and responsibilities required to establish and operate an architecture function within an enterprise.

The intention of dividing the TOGAF specification into these independent parts is to allow for different areas of specialization to be considered in detail and potentially addressed in isolation. Although all parts work together as a whole, it is also feasible to select particular parts for adoption while excluding others. For example, an organization may wish to adopt the ADM process, but elect not to use any of the materials relating to Architecture Capability.

As an open framework, such use is encouraged, particularly in the following situations:

- Organizations that are new to TOGAF and wish to incrementally adopt TOGAF concepts are expected to focus on particular parts of the specification for initial adoption, with other areas tabled for later consideration.
- Organizations that have already deployed architecture frameworks may choose to merge these frameworks with aspects of the TOGAF specification.

1.2 Executive Overview

This section provides an executive overview of enterprise architecture, the basic concepts of what it is (not just another name for IT Architecture), and why it is needed. It provides a summary of the benefits of establishing an enterprise architecture and adopting TOGAF to achieve that.

What is an enterprise?

TOGAF defines ‘‘enterprise’’ as any collection of organizations that has a common set of goals. For example, an enterprise could be a government agency, a whole corporation, a division of a corporation, a single department, or a chain of geographically distant organizations linked together by common ownership.

The term ‘‘enterprise’’ in the context of ‘‘enterprise architecture’’ can be used to denote both an entire enterprise — encompassing all of its information and technology services, processes, and infrastructure — and a specific domain within the enterprise. In both cases, the architecture crosses multiple systems, and multiple functional groups within the enterprise.

Confusion often arises from the evolving nature of the term ‘‘enterprise’’. An extended enterprise nowadays frequently includes partners, suppliers, and customers. If the goal is to integrate an extended enterprise, then the enterprise comprises the partners, suppliers, and customers, as well as internal business units.

The business operating model concept is useful to determine the nature and scope of the enterprise architecture within an organization. Large corporations and government agencies may comprise multiple enterprises, and may develop and maintain a number of independent enterprise architectures to address each one. However, there is often much in common about the information systems in each enterprise, and there is usually great potential for gain in the

use of a common architecture framework. For example, a common framework can provide a basis for the development of an Architecture Repository for the integration and re-use of models, designs, and baseline data.

Why do I need an enterprise architecture?

The purpose of enterprise architecture is to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy.

Today's CEOs know that the effective management and exploitation of information through IT is a key factor to business success, and an indispensable means to achieving competitive advantage. An enterprise architecture addresses this need, by providing a strategic context for the evolution of the IT system in response to the constantly changing needs of the business environment.

Furthermore, a good enterprise architecture enables you to achieve the right balance between IT efficiency and business innovation. It allows individual business units to innovate safely in their pursuit of competitive advantage. At the same time, it ensures the needs of the organization for an integrated IT strategy are met, permitting the closest possible synergy across the extended enterprise.

The advantages that result from a good enterprise architecture bring important business benefits, which are clearly visible in the net profit or loss of a company or organization:

- A more efficient business operation:
  - Lower business operation costs
  - More agile organization
  - Business capabilities shared across the organization
  - Lower change management costs
  - More flexible workforce
  - Improved business productivity
- A more efficient IT operation:
  - Lower software development, support, and maintenance costs
  - Increased portability of applications
  - Improved interoperability and easier system and network management
  - Improved ability to address critical enterprise-wide issues like security
  - Easier upgrade and exchange of system components
- Better return on existing investment, reduced risk for future investment:
  - Reduced complexity in the business and IT
  - Maximum return on investment in existing business and IT infrastructure
  - The flexibility to make, buy, or out-source business and IT solutions
  - Reduced risk overall in new investments and their cost of ownership

- Faster, simpler, and cheaper procurement:
  - — Buying decisions are simpler, because the information governing procurement is readily available in a coherent plan
  - — The procurement process is faster — maximizing procurement speed and flexibility without sacrificing architectural coherence
  - — The ability to procure heterogeneous, multi-vendor open systems
  - — The ability to secure more economic capabilities

What specifically would prompt me to develop an enterprise architecture?

Typically, preparation for business transformation needs or for radical infrastructure changes initiates an enterprise architecture review or development. Often key people identify areas of change required in order for new business goals to be met. Such people are commonly referred to as the ‘‘stakeholders’’ in the change. The role of the architect is to address their concerns by:

- Identifying and refining the requirements that the stakeholders have
- Developing views of the architecture that show how the concerns and requirements are going to be addressed
- Showing the trade-offs that are going to be made in reconciling the potentially conflicting concerns of different stakeholders

Without the enterprise architecture, it is highly unlikely that all the concerns and requirements will be considered and met.

What is an architecture framework?

An architecture framework is a foundational structure, or set of structures, which can be used for developing a broad range of different architectures. It should describe a method for designing a target state of the enterprise in terms of a set of building blocks, and for showing how the building blocks fit together. It should contain a set of tools and provide a common vocabulary. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks.

Why do I need TOGAF as a framework for enterprise architecture?

TOGAF has been developed through the collaborative efforts of over 300 Architecture Forum member companies from some of the world’s leading companies and organizations. Using TOGAF results in enterprise architecture that is consistent, reflects the needs of stakeholders, employs best practice, and gives due consideration both to current requirements and the perceived future needs of the business.

Developing and sustaining an enterprise architecture is a technically complex process which involves many stakeholders and decision processes in the organization. TOGAF plays an important role in standardizing and de-risks the architecture development process. TOGAF provides a best practice framework for adding value, and enables the organization to build workable and economic solutions which address their business issues and needs.

Who would benefit from using TOGAF?

Any organization undertaking, or planning to undertake, the development and implementation of an enterprise architecture for the support of business transformation will benefit from use of TOGAF.

Organizations seeking Boundaryless Information Flow can use TOGAF to define and implement the structures and processes to enable access to integrated information within and between enterprises.

Organizations that design and implement enterprise architectures using TOGAF are assured of a design and a procurement specification that can facilitate an open systems implementation, thus enabling the benefits of open systems with reduced risk.

Chapter 2

Definitions

This extract from TOGAF 9.2 contains the general definitions required for the Level 1 Syllabus, section 3.1.3, learning unit 3. It is primarily intended for TOGAF 9.2 Level 1 training.

2.1 Application

A deployed and operational IT system that supports business functions and services; for example, a payroll. Applications use data and are supported by multiple technology components but are distinct from the technology components that support the application.

2.2 Application Architecture

A description of the structure and interaction of the applications as groups of capabilities that provide key business functions and manage the data assets.

2.3 Architecture

1. A formal description of a system, or a detailed plan of the system at component level, to guide its implementation (source: ISO/IEC 42010: 2007).
2. The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.

2.4 Architecture Building Block (ABB)

A constituent of the architecture model that describes a single aspect of the overall model.

2.5 Architecture Development Method (ADM)

The core of TOGAF. A step-by-step approach to develop and use an enterprise architecture.

2.6 Architecture Domain

The architectural area being considered. There are four architecture domains within TOGAF: business, data, application, and technology.

2.7 Architecture Framework

A conceptual structure used to develop, implement, and sustain an architecture.

2.8 Architecture Principles

A qualitative statement of intent that should be met by the architecture. Has at least a supporting rationale and a measure of importance.

2.9 Architecture Vision

A succinct description of the Target Architecture that describes its business value and the changes to the enterprise that will result from its successful deployment. It serves as an aspirational vision and a boundary for detailed architecture development.

2.10 Baseline

A specification that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development or change and that can be changed only through formal change control procedures or a type of procedure such as configuration management.

2.11 Building Block

Represents a (potentially re-usable) component of business, IT, or architectural capability that can be combined with other building blocks to deliver architectures and solutions.

Building blocks can be defined at various levels of detail, depending on what stage of architecture development has been reached. For instance, at an early stage, a building block can simply consist of a name or an outline description. Later on, a building block may be decomposed into multiple supporting building blocks and may be accompanied by a full specification. Building blocks can relate to ‘‘architectures’’ or ‘‘solutions’’.

2.12 Business Architecture

A description of the structure and interaction between the business strategy, organization, functions, business processes, and information needs.

2.13 Business Governance

Concerned with ensuring that the business processes and policies (and their operation) deliver the business outcomes and adhere to relevant business regulation.

2.14 Capability

An ability that an organization, person, or system possesses. Capabilities are typically expressed in general and high-level terms and typically require a combination of organization, people, processes, and technology to achieve. For example, marketing, customer contact, or outbound telemarketing.

2.15 Concerns

The key interests that are crucially important to the stakeholders in a system, and determine the acceptability of the system. Concerns may pertain to any aspect of the system’s functioning, development, or operation, including considerations such as performance, reliability, security, distribution, and evolvability.

2.16 Constraint

An external factor that prevents an organization from pursuing particular approaches to meet its goals. For example, customer data is not harmonized within the organization, regionally or nationally, constraining the organization’s ability to offer effective customer service.

2.17 Data Architecture

A description of the structure and interaction of the enterprise's major types and sources of data, logical data assets, physical data assets, and data management resources.

2.18 Deliverable

An architectural work product that is contractually specified and in turn formally reviewed, agreed, and signed off by the stakeholders. Deliverables represent the output of projects and those deliverables that are in documentation form will typically be archived at completion of a project, or transitioned into an Architecture Repository as a reference model, standard, or snapshot of the Architecture Landscape at a point in time.

2.19 Enterprise

The highest level (typically) of description of an organization and typically covers all missions and functions. An enterprise will often span multiple organizations.

2.20 Foundation Architecture

Generic building blocks, their inter-relationships with other building blocks, combined with the principles and guidelines that provide a foundation on which more specific architectures can be built.

2.21 Gap

A statement of difference between two states. Used in the context of gap analysis, where the difference between the Baseline and Target Architecture is identified.

2.22 Governance

The discipline of monitoring, managing, and steering a business (or IS/IT landscape) to deliver the business outcome required.

2.23 Information

Any communication or representation of facts, data, or opinions, in any medium or form, including textual, numerical, graphic, cartographic, narrative, or audio-visual forms.

2.24 Information Technology (IT)

1. The lifecycle management of information and related technology used by an organization.
2. An umbrella term that includes all or some of the subject areas relating to the computer industry, such as Business Continuity, Business IT Interface, Business Process Modeling and Management, Communication, Compliance and Legislation, Computers, Content Management, Hardware, Information Management, Internet, Offshoring, Networking, Programming and Software, Professional Issues, Project Management, Security, Standards, Storage, Voice and Data Communications. Various countries and industries employ other umbrella terms to describe this same collection.
3. A term commonly assigned to a department within an organization tasked with provisioning some or all of the domains described in (2) above.
4. Alternate names commonly adopted include Information Services, Information Management, et al.

2.25 Logical

An implementation-independent definition of the architecture, often grouping related physical entities according to their purpose and structure. For example, the products from multiple infrastructure software vendors can all be logically grouped as Java application server platforms.

2.26 Metadata

Data about data, of any sort in any media, that describes the characteristics of an entity.

2.27 Metamodel

A model that describes how and with what the architecture will be described in a structured way.

2.28 Method

A defined, repeatable approach to address a particular type of problem.

See also Section 2.29.

2.29 Methodology

A defined, repeatable series of steps to address a particular type of problem, which typically centers on a defined process, but may also include definition of content.

2.30 Model

A representation of a subject of interest. A model provides a smaller scale, simplified, and/or abstract representation of the subject matter. A model is constructed as a ‘‘means to an end’’. In the context of enterprise architecture, the subject matter is a whole or part of the enterprise and the end is the ability to construct ‘‘views’’ that address the concerns of particular stakeholders; i.e., their ‘‘viewpoints’’ in relation to the subject matter.

2.31 Modeling

A technique through construction of models which enables a subject to be represented in a form that enables reasoning, insight, and clarity concerning the essence of the subject matter.

2.32 Objective

A time-bounded milestone for an organization used to demonstrate progress towards a goal; for example, ‘‘Increase Capacity Utilization by 30% by the end of 2009 to support the planned increase in market share’’.

2.33 Physical

A description of a real-world entity. Physical elements in an enterprise architecture may still be considerably abstracted from Solution Architecture, design, or implementation views.

2.34 Reference Model (RM)

A reference model is an abstract framework for understanding significant relationships among the entities of [an] environment, and for the development of consistent standards or specifications supporting that environment. A reference model is based on a small number of unifying concepts and may be used as a basis for education and explaining standards to a non-specialist. A reference model is not directly tied to any standards, technologies, or other concrete implementation details, but it does seek to provide common semantics that can be used unambiguously across and between different implementations.

2.35 Repository

A system that manages all of the data of an enterprise, including data and process models and other enterprise information. Hence, the data in a repository is much more extensive than that in a data dictionary, which generally defines only the data making up a database.

2.36 Requirement

A statement of need that must be met by a particular architecture or work package.

2.37 Solution Architecture

A description of a discrete and focused business operation or activity and how IS/IT supports that operation. A Solution Architecture typically applies to a single project or project release, assisting in the translation of requirements into a solution vision, high-level business and/or IT system specifications, and a portfolio of implementation tasks.

2.38 Solution Building Block (SBB)

A candidate solution which conforms to the specification of an Architecture Building Block (ABB).

2.39 Stakeholder

An individual, team, or organization (or classes thereof) with interests in, or concerns relative to, the outcome of the architecture. Different stakeholders with different roles will have different concerns.

2.40 Strategic Architecture

A summary formal description of the enterprise, providing an organizing framework for operational and change activity, and an executive-level, long-term view for direction setting.

2.41 Target Architecture

The description of a future state of the architecture being developed for an organization. There may be several future states developed as a roadmap to show the evolution of the architecture to a target state.

2.42 Technology Architecture

A description of the structure and interaction of the platform services, and logical and physical technology components.

2.43 Transition Architecture

A formal description of one state of the architecture at an architecturally significant point in time. One or more Transition Architectures may be used to describe the progression in time from the Baseline to the Target Architecture.

2.44 View

The representation of a related set of concerns. A view is what is seen from a viewpoint. An architecture view may be represented by a model to demonstrate to stakeholders their areas of interest in the architecture. A view does not have to be visual or graphical in nature.

2.45 Viewpoint

A definition of the perspective from which a view is taken. It is a specification of the conventions for constructing and using a view (often by means of an appropriate schema or template). A view is what you see; a viewpoint is where you are looking from — the vantage point or perspective that determines what you see.

Chapter 3

Architecture Principles

See the course slides for when to consult this handout. This handout is derived from Chapter 23 of TOGAF 9.2. It is intended to be used for TOGAF 9.2 Level 2 training. It may also be used to introduce Principles at TOGAF 9.2 Level 1 training.

3.1 Introduction

Principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission.

In their turn, principles may be just one element in a structured set of ideas that collectively define and guide the organization, from values through to actions and results.

Depending on the organization, principles may be established within different domains and at different levels. Two key domains inform the development and utilization of architecture:

- **Enterprise** principles provide a basis for decision-making throughout an enterprise, and inform how the organization sets about fulfilling its mission. Such principles are commonly found as a means of harmonizing decision-making across an organization. In particular, they are a key element in a successful architecture governance strategy.

  Within the broad domain of enterprise principles, it is common to have subsidiary principles within a business or organizational unit. Examples include IT, HR, domestic operations, or overseas operations. These principles provide a basis for decision-making within the subsidiary domain and will inform architecture development within the domain. Care must be taken to ensure that the principles used to inform architecture development align to the organizational context of the Architecture Capability.

- **Architecture** principles are a set of principles that relate to architecture work. They reflect a level of consensus across the enterprise, and embody the spirit and thinking of existing enterprise principles. Architecture principles govern the architecture process, affecting the development, maintenance, and use of the enterprise architecture.

It is common to have sets of principles form a hierarchy, in that segment principles will be informed by, and elaborate on, the principles at the enterprise level. Architecture principles will be informed and constrained by enterprise principles.

Architecture principles may restate other enterprise guidance in terms and form that effectively guide architecture development.

The remainder of this section deals exclusively with architecture principles.

3.2 Characteristics of Architecture Principles

Architecture principles define the underlying general rules and guidelines for the use and deployment of all IT resources and assets across the enterprise. They reflect a level of consensus among the various elements of the enterprise, and form the basis for making future IT decisions.

Each architecture principle should be clearly related back to the business objectives and key architecture drivers.

3.3 Components of Architecture Principles

It is useful to have a standard way of defining principles. In addition to a definition statement, each principle should have associated rationale and implications statements, both to promote understanding and acceptance of the principles themselves, and to support the use of the principles in explaining and justifying why specific decisions are made.

A recommended template is given in Table 3-1.

Name	Should both represent the essence of the rule as well as be easy to remember. Specific technology platforms should not be mentioned in the name or statement of a principle. Avoid ambiguous words in the Name and in the Statement such as: ‘‘support’’, ‘‘open’’, ‘‘consider’’, and for lack of good measure the word ‘‘avoid’’, itself, be careful with ‘‘manage(ment)’’, and look for unnecessary adjectives and adverbs (fluff).
Statement	Should succinctly and unambiguously communicate the fundamental rule. For the most part, the principles statements for managing information are similar from one organization to the next. It is vital that the principles statement be unambiguous.
Rationale	Should highlight the business benefits of adhering to the principle, using business terminology. Point to the similarity of information and technology principles to the principles governing business operations. Also describe the relationship to other principles, and the intentions regarding a balanced interpretation. Describe situations where one principle would be given precedence or carry more weight than another for making a decision.
Implications	Should highlight the requirements, both for the business and IT, for carrying out the principle — in terms of resources, costs, and activities/tasks. It will often be apparent that current systems, standards, or practices would be incongruent with the principle upon adoption. The impact to the business and consequences of adopting a principle should be clearly stated. The reader should readily discern the answer to: ‘‘How does this affect me?’’ It is important not to oversimplify, trivialize, or judge the merit of the impact. Some of the implications will be identified as potential impacts only, and may be speculative rather than fully analyzed.

Table 3-1 Recommended Format for Defining Principles

An example set of architecture principles following this template is given in Section 3.6.

3.4 Developing Architecture Principles

Architecture principles are typically developed by the enterprise architects, in conjunction with the key stakeholders, and are approved by the Architecture Board.

Architecture principles will be informed by principles at the enterprise level, if they exist.

Architecture principles must be clearly traceable and clearly articulated to guide decision-making. They are chosen so as to ensure alignment of the architecture and implementation of the Target Architecture with business strategies and visions.

Specifically, the development of architecture principles is typically influenced by the following:

- **Enterprise mission and plans**: the mission, plans, and organizational infrastructure of the enterprise.
- **Enterprise strategic initiatives**: the characteristics of the enterprise — its strengths, weaknesses, opportunities, and threats — and its current enterprise-wide initiatives (such as process improvement and quality management).
- **External constraints**: market factors (time-to-market imperatives, customer expectations, etc.); existing and potential legislation.
- **Current systems and technology**: the set of information resources deployed within the enterprise, including systems documentation, equipment inventories, network configuration diagrams, policies, and procedures.
- **Emerging industry trends**: predictions about economic, political, technical, and market factors that influence the enterprise environment.

3.4.1 Qualities of Principles

Merely having a written statement that is called a principle does not mean that the principle is good, even if everyone agrees with it.

A good set of principles will be founded in the beliefs and values of the organization and expressed in language that the business understands and uses. Principles should be few in number, future-oriented, and endorsed and championed by senior management. They provide a firm foundation for making architecture and planning decisions, framing policies, procedures, and standards, and supporting resolution of contradictory situations. A poor set of principles will quickly become disused, and the resultant architectures, policies, and standards will appear arbitrary or self-serving, and thus lack credibility. Essentially, principles drive behavior.

There are five criteria that distinguish a good set of principles:

- **Understandable**: the underlying tenets can be quickly grasped and understood by individuals throughout the organization. The intention of the principle is clear and unambiguous, so that violations, whether intentional or not, are minimized.
- **Robust**: enable good quality decisions about architectures and plans to be made, and enforceable policies and standards to be created. Each principle should be sufficiently definitive and precise to support consistent decision-making in complex, potentially controversial situations.
- **Complete**: every potentially important principle governing the management of information and technology for the organization is defined. The principles cover every situation perceived.

- **Consistent**: strict adherence to one principle may require a loose interpretation of another principle. The set of principles must be expressed in a way that allows a balance of interpretations. Principles should not be contradictory to the point where adhering to one principle would violate the spirit of another. Every word in a principle statement should be carefully chosen to allow consistent yet flexible interpretation.
- **Stable**: principles should be enduring, yet able to accommodate changes. An amendment process should be established for adding, removing, or altering principles after they are ratified initially.

3.5 Applying Architecture Principles

Architecture principles are used to capture the fundamental truths about how the enterprise will use and deploy IT resources and assets. The principles are used in a number of different ways:

1. To provide a framework within which the enterprise can start to make conscious decisions about enterprise architecture and projects that implement the target enterprise architecture
2. As a guide to establishing relevant evaluation criteria, thus exerting strong influence on the selection of products, solutions, or solution architectures in the later stages of managing compliance to the enterprise architecture
3. As drivers for defining the functional requirements of the architecture
4. As an input to assessing both existing implementations and the strategic portfolio, for compliance with the defined architectures; these assessments will provide valuable insights into the transition activities needed to implement an architecture, in support of business goals and priorities
5. The Rationale statements within an Architecture Principle highlight the business value of implementations consistent with the principle and provide guidance for difficult decisions with conflicting drivers or objectives
6. The Implications statements within an Architecture Principle provide an outline of the key tasks, resources, and potential costs to the enterprise of following the principle; they also provide valuable inputs to future transition initiative and planning activities
7. Support the architecture governance activities in terms of:
    - — Providing a ‘‘back-stop’’ for the standard Architecture Compliance assessments where some interpretation is allowed or required
    - — Supporting the decision to initiate a dispensation request where the implications of a particular architecture amendment cannot be resolved within local operating procedure

Principles are inter-related, and need to be applied as a set.

Principles will sometimes compete; for example, the principles of ‘‘accessibility’’ and ‘‘security’’ tend towards conflicting decisions. Each principle must be considered in the context of ‘‘all other things being equal’’.

At times a decision will be required as to which principle will take precedence on a particular issue. The rationale for such decisions should always be documented.

A common reaction on first reading of a principle is ‘‘this is obvious and does not need to be documented’’. The fact that a principle seems self-evident does not mean that the guidance in a

principle is followed. Having principles that appear obvious helps ensure that decisions actually follow the desired outcome.

Although specific penalties are not prescribed in a declaration of principles, violations of principles generally cause operational problems and inhibit the ability of the organization to fulfil its mission.

3.6 Example Set of Architecture Principles

Too many principles can reduce the flexibility of the architecture. Many organizations prefer to define only high-level principles, and to limit the number to between 10 and 20.

The following example illustrates both the typical content of a set of architecture principles, and the recommended format for defining them, as explained above.

3.6.1 Business Principles

Principle 1: Primary of Principles

Statement: These principles of information management apply to all organizations within the enterprise.

Rationale: The only way we can provide a consistent and measurable level of quality information to decision-makers is if all organizations abide by the principles.

Implications:

- Without this principle, exclusions, favoritism, and inconsistency would rapidly undermine the management of information.
- Information management initiatives will not begin until they are examined for compliance with the principles.
- A conflict with a principle will be resolved by changing the framework of the initiative.

Principle 2: Maximize Benefit to the Enterprise

Statement: Information management decisions are made to provide maximum benefit to the enterprise as a whole.

Rationale: This principle embodies ‘‘service above self’’. Decisions made from an enterprise-wide perspective have greater long-term value than decisions made from any particular organizational perspective. Maximum return on investment requires information management decisions to adhere to enterprise-wide drivers and priorities. No minority group will detract from the benefit of the whole. However, this principle will not preclude any minority group from getting its job done.

Implications:

- Achieving maximum enterprise-wide benefit will require changes in the way we plan and manage information. Technology alone will not bring about this change.
- Some organizations may have to concede their own preferences for the greater benefit of the entire enterprise.

- Application development priorities must be established by the entire enterprise for the entire enterprise.
- Applications components should be shared across organizational boundaries.
- Information management initiatives should be conducted in accordance with the enterprise plan. Individual organizations should pursue information management initiatives which conform to the blueprints and priorities established by the enterprise. We will change the plan as we need to.
- As needs arise, priorities must be adjusted. A forum with comprehensive enterprise representation should make these decisions.

Principle 3: Information Management is Everybody's Business

Statement: All organizations in the enterprise participate in information management decisions needed to accomplish business objectives.

Rationale: Information users are the key stakeholders, or customers, in the application of technology to address a business need. In order to ensure information management is aligned with the business, all organizations in the enterprise must be involved in all aspects of the information environment. The business experts from across the enterprise and the technical staff responsible for developing and sustaining the information environment need to come together as a team to jointly define the goals and objectives of IT.

Implications:
- To operate as a team, every stakeholder, or customer, will need to accept responsibility for developing the information environment.
- Commitment of resources will be required to implement this principle.

Principle 4: Business Continuity

Statement: Enterprise operations are maintained in spite of system interruptions.

Rationale: As system operations become more pervasive, we become more dependent on them; therefore, we must consider the reliability of such systems throughout their design and use. Business premises throughout the enterprise must be provided with the capability to continue their business functions regardless of external events. Hardware failure, natural disasters, and data corruption should not be allowed to disrupt or stop enterprise activities. The enterprise business functions must be capable of operating on alternative information delivery mechanisms.

Implications:
- Dependency on shared system applications mandates that the risks of business interruption must be established in advance and managed. Management includes but is not limited to periodic reviews, testing for vulnerability and exposure, or designing mission-critical services to ensure business function continuity through redundant or alternative capabilities.
- Recoverability, redundancy, and maintainability should be addressed at the time of design.
- Applications must be assessed for criticality and impact on the enterprise mission, in order to determine what level of continuity is required and what corresponding recovery plan is necessary.

Principle 5: Common Use Applications

Statement: Development of applications used across the enterprise is preferred over the development of similar or duplicative applications which are only provided to a particular organization.

Rationale: Duplicative capability is expensive and proliferates conflicting data.

Implications:

- Organizations which depend on a capability which does not serve the entire enterprise must change over to the replacement enterprise-wide capability. This will require establishment of and adherence to a policy requiring this.
- Organizations will not be allowed to develop capabilities for their own use which are similar/duplicative of enterprise-wide capabilities. In this way, expenditures of scarce resources to develop essentially the same capability in marginally different ways will be reduced.
- Data and information used to support enterprise decision-making will be standardized to a much greater extent than previously. This is because the smaller, organizational capabilities which produced different data (which was not shared among other organizations) will be replaced by enterprise-wide capabilities. The impetus for adding to the set of enterprise-wide capabilities may well come from an organization making a convincing case for the value of the data/information previously produced by its organizational capability, but the resulting capability will become part of the enterprise-wide system, and the data it produces will be shared across the enterprise.

Principle 6: Service Orientation

Statement: The architecture is based on a design of services which mirror real-world business activities comprising the enterprise (or inter-enterprise) business processes.

Rationale: Service orientation delivers enterprise agility and Boundaryless Information Flow.

Implications:

- Service representation utilizes business descriptions to provide context (i.e., business process, goal, rule, policy, service interface, and service component) and implements services using service orchestration.
- Service orientation places unique requirements on the infrastructure, and implementations should use open standards to realize interoperability and location transparency.
- Implementations are environment-specific; they are constrained or enabled by context and must be described within that context.
- Strong governance of service representation and implementation is required.
- A ‘‘Litmus Test’’, which determines a ‘‘good service’’, is required.

Principle 7: Compliance with Law

Statement: Enterprise information management processes comply with all relevant laws, policies, and regulations.

Rationale: Enterprise policy is to abide by laws, policies, and regulations. This will not preclude business process improvements that lead to changes in policies and regulations.

Implications:

- The enterprise must be mindful to comply with laws, regulations, and external policies regarding the collection, retention, and management of data.
- Education and access to the rules. Efficiency, need, and common sense are not the only drivers. Changes in the law and changes in regulations may drive changes in our processes or applications.

Principle 8: IT Responsibility

Statement: The IT organization is responsible for owning and implementing IT processes and infrastructure that enable solutions to meet user-defined requirements for functionality, service levels, cost, and delivery timing.

Rationale: Effectively align expectations with capabilities and costs so that all projects are cost-effective. Efficient and effective solutions have reasonable costs and clear benefits.

Implications:

- A process must be created to prioritize projects.
- The IT function must define processes to manage business unit expectations.
- Data, application, and technology models must be created to enable integrated quality solutions and to maximize results.

Principle 9: Protection of Intellectual Property

Statement: The enterprise's Intellectual Property (IP) must be protected. This protection must be reflected in the IT architecture, implementation, and governance processes.

Rationale: A major part of an enterprise's IP is hosted in the IT domain.

Implications:

- While protection of IP assets is everybody's business, much of the actual protection is implemented in the IT domain. Even trust in non-IT processes can be managed by IT processes (email, mandatory notes, etc.).
- A security policy, governing human and IT actors, will be required that can substantially improve protection of IP. This must be capable of both avoiding compromises and reducing liabilities.
- Resources on such policies can be found at the SANS Institute (refer to www.sans.org/newlook/home.php).

3.6.2 Data Principles

Principle 10: Data is an Asset

Statement: Data is an asset that has value to the enterprise and is managed accordingly.

Rationale: Data is a valuable corporate resource; it has real, measurable value. In simple terms, the purpose of data is to aid decision-making. Accurate, timely data is critical to accurate, timely decisions. Most corporate assets are carefully managed, and data is no exception. Data is the foundation of our decision-making, so we must also carefully manage data to ensure that we know where it is, can rely upon its accuracy, and can obtain it when and where we need it.

Implications:

- This is one of three closely-related principles regarding data: data is an asset; data is shared; and data is easily accessible. The implication is that there is an education task to ensure that all organizations within the enterprise understand the relationship between value of data, sharing of data, and accessibility to data.
- Stewards must have the authority and means to manage the data for which they are accountable.
- We must make the cultural transition from ‘‘data ownership’’ thinking to ‘‘data stewardship’’ thinking.
- The role of data steward is critical because obsolete, incorrect, or inconsistent data could be passed to enterprise personnel and adversely affect decisions across the enterprise.
- Part of the role of data steward, who manages the data, is to ensure data quality. Procedures must be developed and used to prevent and correct errors in the information and to improve those processes that produce flawed information. Data quality will need to be measured and steps taken to improve data quality — it is probable that policy and procedures will need to be developed for this as well.
- A forum with comprehensive enterprise-wide representation should decide on process changes suggested by the steward.
- Since data is an asset of value to the entire enterprise, data stewards accountable for properly managing the data must be assigned at the enterprise level.

Principle 11: Data is Shared

Statement: Users have access to the data necessary to perform their duties; therefore, data is shared across enterprise functions and organizations.

Rationale: Timely access to accurate data is essential to improving the quality and efficiency of enterprise decision-making. It is less costly to maintain timely, accurate data in a single application, and then share it, than it is to maintain duplicative data in multiple applications. The enterprise holds a wealth of data, but it is stored in hundreds of incompatible stovepipe databases. The speed of data collection, creation, transfer, and assimilation is driven by the ability of the organization to efficiently share these islands of data across the organization.

Shared data will result in improved decisions since we will rely on fewer (ultimately one virtual) sources of more accurate and timely managed data for

all of our decision-making. Electronically shared data will result in increased efficiency when existing data entities can be used, without re-keying, to create new entities.

Implications:

- This is one of three closely-related principles regarding data: data is an asset; data is shared; and data is easily accessible. The implication is that there is an education task to ensure that all organizations within the enterprise understand the relationship between value of data, sharing of data, and accessibility to data.
- To enable data sharing we must develop and abide by a common set of policies, procedures, and standards governing data management and access for both the short and the long term.
- For the short term, to preserve our significant investment in legacy systems, we must invest in software capable of migrating legacy system data into a shared data environment.
- We will also need to develop standard data models, data elements, and other metadata that defines this shared environment and develop a repository system for storing this metadata to make it accessible.
- For the long term, as legacy systems are replaced, we must adopt and enforce common data access policies and guidelines for new application developers to ensure that data in new applications remains available to the shared environment and that data in the shared environment can continue to be used by the new applications.
- For both the short term and the long term we must adopt common methods and tools for creating, maintaining, and accessing the data shared across the enterprise.
- Data sharing will require a significant cultural change.
- This principle of data sharing will continually ‘‘bump up against’’ the principle of data security. Under no circumstances will the data sharing principle cause confidential data to be compromised.
- Data made available for sharing will have to be relied upon by all users to execute their respective tasks. This will ensure that only the most accurate and timely data is relied upon for decision-making. Shared data will become the enterprise-wide ‘‘virtual single source’’ of data.

Principle 12: Data is Accessible

Statement: Data is accessible for users to perform their functions.

Rationale: Wide access to data leads to efficiency and effectiveness in decision-making, and affords timely response to information requests and service delivery. Using information must be considered from an enterprise perspective to allow access by a wide variety of users. Staff time is saved and consistency of data is improved.

Implications:

- This is one of three closely-related principles regarding data: data is an asset; data is shared; and data is easily accessible. The implication is that there is an education task to ensure that all organizations within the enterprise understand the relationship between value of data, sharing of data, and accessibility to data.

- Accessibility involves the ease with which users obtain information.
- The way information is accessed and displayed must be sufficiently adaptable to meet a wide range of enterprise users and their corresponding methods of access.
- Access to data does not constitute understanding of the data. Personnel should take caution not to misinterpret information.
- Access to data does not necessarily grant the user access rights to modify or disclose the data. This will require an education process and a change in the organizational culture, which currently supports a belief in ''ownership'' of data by functional units.

Principle 13: Data Trustee

Statement: Each data element has a trustee accountable for data quality.

Rationale: One of the benefits of an architected environment is the ability to share data (e.g., text, video, sound, etc.) across the enterprise. As the degree of data sharing grows and business units rely upon common information, it becomes essential that only the data trustee makes decisions about the content of data. Since data can lose its integrity when it is entered multiple times, the data trustee will have sole responsibility for data entry which eliminates redundant human effort and data storage resources.

Note: A trustee is different than a steward — a trustee is responsible for accuracy and currency of the data, while responsibilities of a steward may be broader and include data standardization and definition tasks.

Implications:

- Real trusteeship dissolves the data ''ownership'' issues and allows the data to be available to meet all users' needs. This implies that a cultural change from data ''ownership'' to data ''trusteeship'' may be required.
- The data trustee will be responsible for meeting quality requirements levied upon the data for which the trustee is accountable.
- It is essential that the trustee has the ability to provide user confidence in the data based upon attributes such as ''data source''.
- It is essential to identify the true source of the data in order that the data authority can be assigned this trustee responsibility. This does not mean that classified sources will be revealed nor does it mean the source will be the trustee.
- Information should be captured electronically once and immediately validated as close to the source as possible. Quality control measures must be implemented to ensure the integrity of the data.
- As a result of sharing data across the enterprise, the trustee is accountable and responsible for the accuracy and currency of their designated data element(s) and, subsequently, must then recognize the importance of this trusteeship responsibility.

Principle 14: Common Vocabulary and Data Definitions

Statement: Data is defined consistently throughout the enterprise, and the definitions are understandable and available to all users.

Rationale: The data that will be used in the development of applications must have a common definition throughout the Headquarters to enable sharing of data. A common vocabulary will facilitate communications and enable dialog to be effective. In addition, it is required to interface systems and exchange data.

Implications:

- We are lulled into thinking that this issue is adequately addressed because there are people with ''data administration'' job titles and forums with charters implying responsibility. Significant additional energy and resources must be committed to this task. It is key to the success of efforts to improve the information environment. This is separate from but related to the issue of data element definition, which is addressed by a broad community — this is more like a common vocabulary and definition.
- The enterprise must establish the initial common vocabulary for the business. The definitions will be used uniformly throughout the enterprise.
- Whenever a new data definition is required, the definition effort will be co-ordinated and reconciled with the corporate ''glossary'' of data descriptions. The enterprise data administrator will provide this co-ordination.
- Ambiguities resulting from multiple parochial definitions of data must give way to accepted enterprise-wide definitions and understanding.
- Multiple data standardization initiatives need to be co-ordinated.
- Functional data administration responsibilities must be assigned.

Principle 15: Data Security

Statement: Data is protected from unauthorized use and disclosure. In addition to the traditional aspects of national security classification, this includes, but is not limited to, protection of pre-decisional, sensitive, source selection-sensitive, and proprietary information.

Rationale: Open sharing of information and the release of information via relevant legislation must be balanced against the need to restrict the availability of classified, proprietary, and sensitive information.

Existing laws and regulations require the safeguarding of national security and the privacy of data, while permitting free and open access. Pre-decisional (work-in-progress, not yet authorized for release) information must be protected to avoid unwarranted speculation, misinterpretation, and inappropriate use.

Implications:

- Aggregation of data, both classified and not, will create a large target requiring review and de-classification procedures to maintain appropriate control. Data owners and/or functional users must determine whether the aggregation results in an increased classification level. We will need appropriate policy and procedures to handle this review and de-classification. Access to information based on a need-to-know policy will force regular reviews of the body of information.

- The current practice of having separate systems to contain different classifications needs to be rethought. Is there a software solution to separating classified and unclassified data? The current hardware solution is unwieldy, inefficient, and costly. It is more expensive to manage unclassified data on a classified system. Currently, the only way to combine the two is to place the unclassified data on the classified system, where it must remain.
- In order to adequately provide access to open information while maintaining secure information, security needs must be identified and developed at the data level, not the application level.
- Data security safeguards can be put in place to restrict access to "view only'', or ''never see''. Sensitivity labeling for access to pre-decisional, decisional, classified, sensitive, or proprietary information must be determined.
- Security must be designed into data elements from the beginning; it cannot be added later. Systems, data, and technologies must be protected from unauthorized access and manipulation. Headquarters information must be safeguarded against inadvertent or unauthorized alteration, sabotage, disaster, or disclosure.
- Need new policies on managing duration of protection for pre-decisional information and other works-in-progress, in consideration of content freshness.

3.6.3 Application Principles

Principle 16: Technology Independence

Statement: Applications are independent of specific technology choices and therefore can operate on a variety of technology platforms.

Rationale: Independence of applications from the underlying technology allows applications to be developed, upgraded, and operated in the most cost-effective and timely way. Otherwise technology, which is subject to continual obsolescence and vendor dependence, becomes the driver rather than the user requirements themselves.

Realizing that every decision made with respect to IT makes us dependent on that technology, the intent of this principle is to ensure that Application Software is not dependent on specific hardware and operating systems software.

Implications:

- This principle will require standards which support portability.
- For Commercial Off-The-Shelf (COTS) and Government Off-The-Shelf (GOTS) applications, there may be limited current choices, as many of these applications are technology and platform-dependent.
- Subsystem interfaces will need to be developed to enable legacy applications to interoperate with applications and operating environments developed under the enterprise architecture.
- Middleware should be used to decouple applications from specific software solutions.

- As an example, this principle could lead to use of Java, and future Java-like protocols, which give a high degree of priority to platform-independence.

Principle 17: Ease-of-Use

Statement: Applications are easy to use. The underlying technology is transparent to users, so they can concentrate on tasks at hand.

Rationale: The more a user has to understand the underlying technology, the less productive that user is. Ease-of-use is a positive incentive for use of applications. It encourages users to work within the integrated information environment instead of developing isolated systems to accomplish the task outside of the enterprise's integrated information environment. Most of the knowledge required to operate one system will be similar to others. Training is kept to a minimum, and the risk of using a system improperly is low.

Using an application should be as intuitive as driving a different car.

Implications:

- Applications will be required to have a common ''look-and-feel'' and support ergonomic requirements. Hence, the common look-and-feel standard must be designed and usability test criteria must be developed.
- Guidelines for user interfaces should not be constrained by narrow assumptions about user location, language, systems training, or physical capability. Factors such as linguistics, customer physical infirmities (visual acuity, ability to use keyboard/mouse), and proficiency in the use of technology have broad ramifications in determining the ease-of-use of an application.

3.6.4 Technology Principles

Principle 18: Requirements-Based Change

Statement: Only in response to business needs are changes to applications and technology made.

Rationale: This principle will foster an atmosphere where the information environment changes in response to the needs of the business, rather than having the business change in response to IT changes. This is to ensure that the purpose of the information support — the transaction of business — is the basis for any proposed change. Unintended effects on business due to IT changes will be minimized. A change in technology may provide an opportunity to improve the business process and, hence, change business needs.

Implications:

- Changes in implementation will follow full examination of the proposed changes using the enterprise architecture.
- We don't fund a technical improvement or system development unless a documented business need exists.
- Change management processes conforming to this principle will be developed and implemented.
- This principle may bump up against the responsive change principle. We must ensure the requirements documentation process does not hinder responsive change to meet legitimate business needs. The purpose of this principle is to keep us focused on business, not technology needs —

responsive change is also a business need.

Principle 19: Responsive Change Management

Statement: Changes to the enterprise information environment are implemented in a timely manner.

Rationale: If people are to be expected to work within the enterprise information environment, that information environment must be responsive to their needs.

Implications:

- We have to develop processes for managing and implementing change that do not create delays.
- A user who feels a need for change will need to connect with a ‘‘business expert’’ to facilitate explanation and implementation of that need.
- If we are going to make changes, we must keep the architectures updated.
- Adopting this principle might require additional resources.
- This will conflict with other principles (e.g., maximum enterprise-wide benefit, enterprise-wide applications, etc.).

Principle 20: Control Technical Diversity

Statement: Technological diversity is controlled to minimize the non-trivial cost of maintaining expertise in and connectivity between multiple processing environments.

Rationale: There is a real, non-trivial cost of infrastructure required to support alternative technologies for processing environments. There are further infrastructure costs incurred to keep multiple processor constructs interconnected and maintained.

Limiting the number of supported components will simplify maintainability and reduce costs.

The business advantages of minimum technical diversity include: standard packaging of components; predictable implementation impact; predictable valuations and returns; redefined testing; utility status; and increased flexibility to accommodate technological advancements. Common technology across the enterprise brings the benefits of economies of scale to the enterprise. Technical administration and support costs are better controlled when limited resources can focus on this shared set of technology.

Implications:

- Policies, standards, and procedures that govern acquisition of technology must be tied directly to this principle.
- Technology choices will be constrained by the choices available within the technology blueprint. Procedures for augmenting the acceptable technology set to meet evolving requirements will have to be developed and put in place.
- We are not freezing our technology baseline. We welcome technology advances and will change the technology blueprint when compatibility with the current infrastructure, improvement in operational efficiency, or a required capability has been demonstrated.

Principle 21: Interoperability

Statement: Software and hardware should conform to defined standards that promote interoperability for data, applications, and technology.

Rationale: Standards help ensure consistency, thus improving the ability to manage systems and improve user satisfaction, and protect existing IT investments, thus maximizing return on investment and reducing costs. Standards for interoperability additionally help ensure support from multiple vendors for their products, and facilitate supply chain integration.

Implications:

- Interoperability standards and industry standards will be followed unless there is a compelling business reason to implement a non-standard solution.
- A process for setting standards, reviewing and revising them periodically, and granting exceptions must be established.
- The existing IT platforms must be identified and documented.

Chapter 4

Stakeholder Management

This handout accompanies the Stakeholder Management module. This is extracted from Chapter 24 of TOGAF 9.2 It is primarily intended to be used for TOGAF 9.2 Level 2 training.

4.1 Introduction

Stakeholder Management is an important discipline that successful architecture practitioners can use to win support from others. It helps them ensure that their projects succeed where others fail.

The benefits of successful Stakeholder Management are that:

- The most powerful stakeholders can be identified early and their input can then be used to shape the architecture; this ensures their support and improves the quality of the models produced.
- Support from the more powerful stakeholders will help the engagement win more resource, thus making the architecture engagement more likely to succeed.
- By communicating with stakeholders early and frequently, the architecture team can ensure that they fully understand the architecture process, and the benefits of enterprise architecture; this means they can support the architecture team more actively when necessary.
- The architecture team can more effectively anticipate likely reactions to the architecture models and reports, and can build into the plan the actions that will be needed to capitalize on positive reaction while avoiding or addressing any negative reactions.
- The architecture team can identify conflicting or competing objectives among stakeholders early and develop a strategy to resolve the issues arising from them.

It is essential in any initiative to identify the individuals and groups within the organization who will contribute to the development of the architecture, identify those that will gain and those that will lose from its introduction, and then develop a strategy for dealing with them.

4.2 Approach to Stakeholder Management

Stakeholder analysis should be used during Phase A (Architecture Vision) to identify the key players in the engagement, and also be updated throughout each phase; different stakeholders may be uncovered as the engagement progresses through into Opportunities & Solutions, Migration Planning, and Architecture Change Management.

Complex architectures are extremely hard to manage, not only in terms of the architecture development process itself, but also in terms of obtaining agreement from the large numbers of stakeholders touched by it.

For example, just as a building architect will create wiring diagrams, floor plans, and elevations to describe different facets of a building to its different stakeholders (electricians, owners, planning officials), so an enterprise architect must create different views of the business, information system, and technology architecture for the stakeholders who have concerns related to these aspects.

TOGAF specifically identifies this issue throughout the ADM through the following concepts:

- Stakeholders
- Concerns
- Views
- Viewpoints

4.3 Steps in the Stakeholder Management Process

The following sections detail recommended Stakeholder Management activity.

4.3.1 Identify Stakeholders

Identify the key stakeholders of the enterprise architecture.

The first task is to brainstorm who the main enterprise architecture stakeholders are. As part of this, think of all the people who are affected by it, who have influence or power over it, or have an interest in its successful or unsuccessful conclusion.

It might include senior executives, project organization roles, client organization roles, system developers, alliance partners, suppliers, IT operations, customers, etc.

When identifying stakeholders there is a danger of concentrating too heavily on the formal structure of an organization as the basis for identification. Informal stakeholder groups may be just as powerful and influential as the formal ones.

Most individuals will belong to more than one stakeholder group, and these groups tend to arise as a result of specific events.

Look at who is impacted by the enterprise architecture project:

- Who gains and who loses from this change?
- Who controls change management of processes?
- Who designs new systems?

- Who will make the decisions?
- Who procures IT systems and who decides what to buy?
- Who controls resources?
- Who has specialist skills the project needs?
- Who has influence?

In particular, influencers need to be identified. These will be well respected and moving up, participate in important meetings and committees (look at meeting minutes), know what's going on in the company, be valued by their peers and superiors, and not necessarily be in any formal position of power.

Although stakeholders may be both organizations and people, ultimately the enterprise architecture team will need to communicate with people. It is the correct individual stakeholders within a stakeholder organization that need to be formally identified.

4.3.1.1 Sample Stakeholder Analysis

A sample stakeholder analysis that distinguishes 22 types of stakeholder, in five broad categories, is shown in Figure 4-1. Any particular architecture project may have more, fewer, or different stakeholders; and they may be grouped into more, fewer, or different categories.

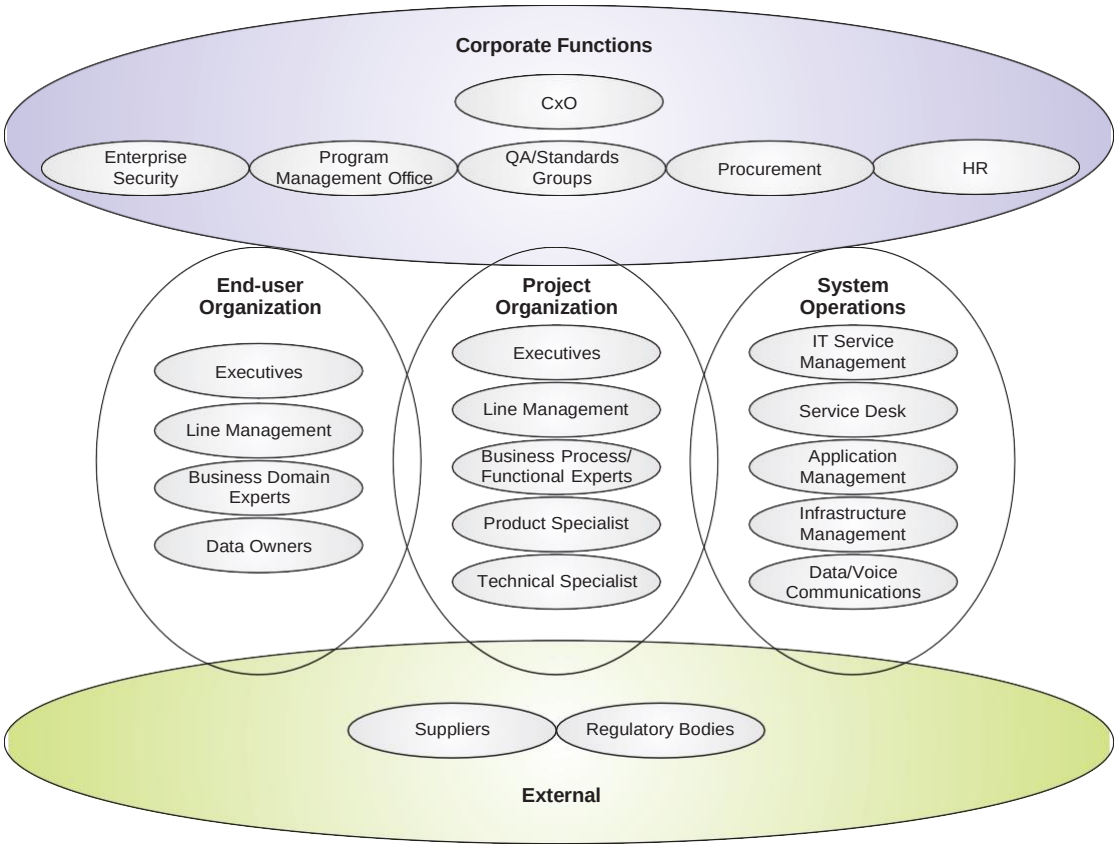

Figure 4-1 Sample Stakeholders and Categories

Consider both the Visible team — those obviously associated with the project/change — and the Invisible team — those who must make a real contribution to the project/change for it to be successful but who are not obviously associated with it (e.g., providers of support services).

4.3.2 Classify Stakeholder Positions

Develop a good understanding of the most important stakeholders and record this analysis for reference and refresh during the project. An example stakeholder analysis is shown in Table 4-1.

Stakeholder Group	Stakeholder	Ability to Disrupt Change	Current Under-standing	Required Under-standing	Current Commit-ment	Required Commit-ment	Required Support
CIO	John Smith	H	M	H	L	M	H
CFO	Jeff Brown	M	M	M	L	M	M

Table 4-1 Example Stakeholder Analysis

It is also important to assess the readiness of each stakeholder to behave in a supportive manner (i.e., demonstrate commitment to the enterprise architecture initiative).

This can be done by asking a series of questions:

- Is that person ready to change direction and begin moving towards the Target Architecture? If so, how ready?
- Is that person capable of being a credible advocate or agent of the proposed enterprise architecture initiative? If so, how capable?
- How involved is the individual in the enterprise architecture initiative? Are they simply an interested observer, or do they need to be involved in the details?
- Has that person made a contractual commitment to the development of the enterprise architecture, and its role in the governance of the development of the organization?

Then, for each person whose commitment is critical to ensure success, make a judgment as to their current level of commitment and the desired future level of commitment.

4.3.3 Determine Stakeholder Management Approach

The previous steps identified a long list of people and organizations that are affected by the enterprise architecture project.

Some of these may have the power either to block or advance. Some may be interested in what the enterprise architecture initiative is doing; others may not care. This step enables the team to easily see which stakeholders are expected to be blockers or critics, and which stakeholders are likely to be advocates and supporters of the initiative.

Work out stakeholder power, influence, and interest, so as to focus the enterprise architecture engagement on the key individuals. These can be mapped onto a power/interest matrix, which also indicates the strategy to adopt for engaging with them. Figure 4-2 shows an example power grid matrix.

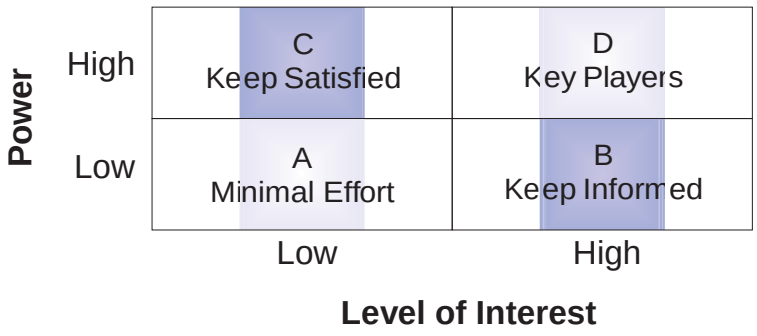

Figure 4-2 Stakeholder Power Grid

4.3.4 Tailor Engagement Deliverables

Identify catalogs, matrices, and diagrams that the architecture engagement needs to produce and validate with each stakeholder group to deliver an effective architecture model.

It is important to pay particular attention to stakeholder interests by defining specific catalogs, matrices, and diagrams that are relevant for a particular enterprise architecture model. This enables the architecture to be communicated to, and understood by, all the stakeholders, and enables them to verify that the enterprise architecture initiative will address their concerns.

4.4 Template Stakeholder Map

The following table provides an example stakeholder map for a TOGAF architecture project which has stakeholders as identified in Figure 4-1.

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
CxO (Corporate Functions); e.g., CEO, CFO, CIO, COO	The high-level drivers, goals, and objectives of the organization, and how these are translated into an effective process and IT architecture to advance the business.	KEEP SATISFIED	Business Footprint diagram Goal/Objective/ Service diagram Organization Decomposition diagram

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Program Management Office (Corporate Functions); e.g., Project Portfolio Managers	Prioritizing, funding, and aligning change activity. An understanding of project content and technical dependencies between projects supports portfolio management decision-making.	KEEP SATISFIED	Requirements catalog Project Context diagram Benefits diagram Business Footprint diagram Application Communication diagram Functional Decomposition diagram
Procurement (Corporate Functions); e.g., Acquirers	Understanding what building blocks of the architecture can be bought, and what constraints (or rules) are relevant to the purchase. Acquirers will shop with multiple vendors looking for the best cost solution while adhering to the constraints (or rules) derived from the architecture, such as standards. The key concern is to make purchasing decisions that fit the architecture.	KEY PLAYERS	Technology Portfolio catalog Technology Standards catalog
Human Resources (HR) (Corporate Functions); e.g., HR Managers, Training & Development Managers	The roles and actors are required to support the architecture and changes to it. The key concern is managing people transitions.	KEEP INFORMED	Organization Decomposition diagram Organization/Actor catalog Location catalog Application and User Location diagram

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Enterprise Security (Corporate Functions); e.g., Corporate Risk Management, Security Officers, IT Security Managers	Ensuring that the information, data, and systems of the organization are available to only those that have permission, and protecting the information, data, and systems from unauthorized tampering.	KEY PLAYERS	Product Lifecycle diagram Data Dissemination diagram Data Security diagram Actor/Role matrix Networked Computing Hardware diagram Communications Engineering diagram
QA/Standards Group (Corporate Functions); e.g., Data Owners, Process Owners, Technical Standards Bodies	Ensuring the consistent governance of the organization's business, data, application, and technology assets.	KEY PLAYERS	Process/Event/ Control/Product catalog Contract/Measure catalog Application Portfolio catalog Interface catalog Technology Standards catalog Technology Portfolio catalog
Executive (End User Organization); e.g., Business Unit Directors, Business Unit CxOs, Business Unit Head of IT/Architecture	The high-level drivers, goals, and objectives of the organization, and how these are translated into an effective process and architecture to advance the business.	KEEP SATISFIED	Business Footprint diagram Goal/Objective/ Service diagram Organization Decomposition diagram Process Flow diagram Application Communication diagram

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Line Management (End User Organization); e.g., Senior Business Managers, Operations Regional Managers, IT Managers	Top-level functions and processes of the organization, and how the key applications support these processes.	KEY PLAYERS	Business Footprint diagram Organization Decomposition diagram Functional Decomposition diagram Process Flow diagram Application Communication diagram Application and User Location diagram
Business Domain Experts (End User Organization); e.g., Business Process Experts, Business/Process Analyst, Process Architect, Process Designer, Functional Managers, Business Analyst	Functional aspects of processes and supporting systems. This can cover the human actors involved in the system, the user processes involved in the system, the functions required to support the processes, and the information required to flow in support of the processes.	KEY PLAYERS	Business Interaction matrix Actor/Role matrix Business Service/ Information diagram Functional Decomposition diagram Product Lifecycle diagram Business Use-case diagram Application Use-case diagram Application Communication diagram Data Entity/Business Function matrix

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
IT Service Management (Systems Operations); e.g., Service Delivery Manager	Ensuring that IT services provided to the organization meet the service levels required by that organization to succeed in business.	KEEP INFORMED	Technology Standards catalog Technology Portfolio catalog Contract/Measure catalog Process/Application Realization diagram Enterprise Manageability diagram
IT Operations — Applications (System Operations); e.g., Application Architecture, System & Software Engineers	Development approach, software modularity and re-use, portability migration, and interoperability.	KEY PLAYERS	Process/Application Realization diagram Application/Data matrix Application Migration diagram Software Engineering diagram Platform decomposition Diagram Networked Computing/ Hardware diagram Software distribution Diagram

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
IT Operations — Infrastructure (System Operations); e.g., Infrastructure Architect, Wintel support, Mid-range support, Operational DBA, Service Desk	Location, modifiability, re-usability, and availability of all components of the system. Ensuring that the appropriate components are developed and deployed within the system in an optimal manner.	KEY PLAYERS	Platform Decomposition diagram Technology Standards catalog Technology Portfolio catalog Enterprise Manageability diagram Networked Computing/ Hardware diagram Processing diagram Environments and Locations diagram
IT Operations — Data/Voice Communications (System Operations); e.g., Network Management	Location, modifiability, re-usability, and availability of communications and networking services. Ensuring that the appropriate communications and networking services are developed and deployed within the system in an optimal manner.	KEY PLAYERS	Communications Engineering diagram
Executive (Project Organization); e.g., Sponsor, Program Manager	On-time, on-budget delivery of a change initiative that will realize expected benefits for the organization.	KEEP INFORMED	Requirements catalog Principles catalog Value Chain diagram Solution Concept diagram Functional Decomposition diagram Application and User Location diagram

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Line Management (Project Organization); e.g., Project Manager	Operationally achieving on-time, on-budget delivery of a change initiative with an agreed scope.	KEEP INFORMED	Application Communication diagram Functional Decomposition diagram Environments and Locations diagram
Business Process/Functional Expert (Project Organization); e.g., Financials FICO Functional Consultant, HR Functional Consultant	Adding more detail to the functional requirements of a change initiative based on experience and interaction with business domain experts in the end-user organization.	KEY PLAYERS	Process Flow diagram Business Use-case diagram Business Service/Information diagram Functional Decomposition diagram Application Communication diagram
Product Specialist (Project Organization); e.g., Portal Product Specialist	Specifying technology product designs in order to meet project requirements and comply with the Architecture Vision of the solution. In a packages and packaged services environment, product expertise can be used to identify product capabilities that can be readily leveraged and can provide guidance on strategies for product customization.	KEY PLAYERS	Software Engineering diagram Application/Data matrix

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Technical Specialist (Project Organization); e.g., Application Architect	Specifying technology product designs in order to meet project requirements and comply with the Architecture Vision of the solution.	KEY PLAYERS	Software Engineering diagram Platform Decomposition diagram Process/Application Realization diagram Application/Data matrix Application Migration diagram
Regulatory Bodies (Outside Services); e.g., Financial Regulator, Industry Regulator	Receipt of the information they need in order to regulate the client organization, and ensuring that their information requirements are properly satisfied. Interested in reporting processes, and the data and applications used to provide regulatory return information.	KEEP SATISFIED	Business Footprint diagram Application Communication diagram
Suppliers (Outside Services); e.g., Alliance Partners, Key Suppliers	Ensuring that their information exchange requirements are met in order that agreed service contracts with the client organizations can be fulfilled.	KEEP SATISFIED	Business Footprint diagram Business Service/Information diagram Application Communication diagram

Chapter 5

Business Scenarios and Business Goals

This handout is extracted from TOGAF 9.2 Chapter 26, supporting Business Scenarios, a method for deriving business requirements for architecture and the implied technical requirements. This handout can be used for TOGAF 9.2 Level 1 and Level 2 training.

5.1 Developing Business Scenarios

5.1.1 General Guidelines

The stakeholders (e.g., business managers, end users) will tell you what they want, but as an architect you must still gain an understanding of the business, so you must know the most important actors in the system. If the stakeholders do not know what they want:

- Take time, observe, and record how they are working today
- Structure information in such a way that it can be used later
- Uncover critical business rules from domain experts
- Stay focused on what needs to be accomplished, and how it is to be accomplished

This effort provides the anchor for a chain of reason from business requirements through to technical solutions. It will pay off later to be diligent and critical at the start.

5.1.2 Questions to Ask for Each Area

The business scenario workshops mentioned above in the Gathering phase are really structured interviews. While there is no single set of appropriate questions to ask in all situations, the following provides some guidance to help business scenario consultants in asking questions.

Identifying, Documenting, and Ranking the Problem

Is the problem described as a statement of *what* needs to be accomplished, like steps in a process, and not *how* (with technology ‘‘push’’)?

If the problem is too specific or a ‘‘how’’:

- Raise a red flag
- Ask “Why do you need to do it that way?” questions

If the problem is too vague or not actionable:

- Raise a red flag
- Ask ‘‘What is it you need to do, or will be able to do if this problem is solved?’’ questions

Ask questions that help to identify where and when the problem exists:

- Where are you experiencing this particular problem? In what business process?
- When do you encounter these issues? During the beginning of the process, the middle, the end?

Ask questions that help to identify the costs of the problem:

- Do you account for the costs associated with this problem? If so, what are they?
- Are there hidden costs? If so, what are they?
- Is the cost of this problem covered in the cost of something else? If so, what and how much?
- Is the problem manifested in terms of poor quality or a perception of an ineffective organization?

Identifying the Business & Technical Environment, and Documenting in Models

Questions to ask about the business environment:

- What key process suffers from the issues? What are the major steps that need to be processed?
- Location/scale of internal business departments?
- Location/scale of external business partners?
- Any specific business rules and regulations related to the situation?

Questions to ask about the current technology environment:

- What technology components are already presupposed to be related to this problem?
- Are there any technology constraints?
- Are there any technology principles that apply?

Identifying and Documenting Objectives

Is the ‘‘what’’ sufficiently backed up with the rationale for ‘‘why’’? If not, ask for measurable rationale in the following areas:

- Return on investment
- Scalability
- Performance needs
- Compliance to standards
- Ease-of-use measures

Identifying Human Actors and their Place in the Business Model

An actor represents anything that interacts with or within the system. This can be a human, or a machine, or a computer program. Actors initiate activity with the system, for example:

- Computer user with the computer
- Phone user with the telephone
- Payroll clerk with the payroll system
- Internet subscriber with the web browser

An actor represents a role that a user plays; i.e., a user is someone playing a role while using the system (e.g., John (user) is a dispatcher (actor)). Each actor uses the system in different ways (otherwise they should be the same actor). Ask about the humans that will be involved, from different viewpoints, such as:

- Developer
- Maintainer
- Operator
- Administrator
- User

Identifying Computer Actors and their Place in the Technology Model

Ask about the computer components likely to be involved, again from different points of view. What must they do?

Documenting Roles, Responsibilities, Measures of Success, Required Scripts

When defining roles, ask questions like:

- What are the main tasks of the actor?
- Will the actor have to read/write/change any information?
- Will the actor have to inform the system about outside changes?
- Does the actor wish to be informed about unexpected changes?

Checking for Fitness-for-Purpose, and refining if necessary

Is there enough information to identify who/what could fulfil the requirement? If not, probe more deeply.

Is there a description of when, and how often, the requirement needs to be addressed? If not, ask about timing.

5.2 Business Scenario Documentation

5.2.1 Textual Documentation

Effective business scenario documentation requires a balance between ensuring that the detail is accessible, and preventing it from overshadowing the results and overwhelming the reader. To this end, the business scenario document should have the main findings in the body of the document and the details in appendices.

In the appendices:

- Capture all the important details about a business scenario:
  - Situation description and rationale
  - All measurements
  - All actor roles and sub-measurements
  - All services required
- Capture the critical steps between actors that address the situation, and sequence the interactions
- Declare relevant information about all actors:
  - Partition the responsibility of the actors
  - List pre-conditions that have to be met prior to proper system functionality
  - Provide technical requirements for the service to be of acceptable quality

In the main body of the business scenario:

- Generalize all the relevant data from the detail in the appendices

5.2.2 Business Scenario Models

- Remember the purpose of using models:
  - Help comprehension
  - Give a starting point to confirm requirements
  - Relate actors and interactions
- Keep drawings clear and neat:
  - Do not put too much into one diagram
  - Simpler diagrams are easier to understand
- Number diagrams for easy reference:
  - Maintain a catalog of the numbers to avoid duplicates

5.3 Guidelines on Goals and Objectives

5.3.1 Importance of Goals

One of the first steps in the development of an architecture is to define the overall goals and objectives for the development. The objectives should be derived from the business goals of the organization, and the way in which IT is seen to contribute to meeting those goals.

Every organization behaves differently in this respect, some seeing IT as the driving force for the enterprise and others seeing IT in a supporting role, simply automating the business processes which already exist. The essential thing is that the architectural objectives should be very closely aligned with the business goals and objectives of the organization.

5.3.2 Importance of SMART Objectives

Not only must goals be stated in general terms, but also specific measures need to be attached to them to make them SMART, as described above.

The amount of effort spent in doing this will lead to greater clarity for the sponsors of the architecture evolution cycle. It will pay back by driving proposed solutions much more closely toward the goals at each step of the cycle. It is extremely helpful for the different stakeholders inside the organization, as well as for suppliers and consultants, to have a clear yardstick for measuring fitness-for-purpose. If done well, the ADM can be used to trace specific decisions back to criteria, and thus yield their justification.

The goals below have been adapted from those given in previous versions of TOGAF. These are categories of goals, each with a list of possible objectives. Each of these objectives should be made SMART with specific measures and metrics for the task. However, since the actual work to be done will be specific to the architecture project concerned, it is not possible to provide a list of generic SMART objectives that will relate to any project.

Instead, we provide here some example SMART objectives.

Example of Making Objectives SMART

Under the general goal heading ‘‘Improve User Productivity’’ below, there is an objective to provide a ‘‘Consistent User Interface’’ and it is described as follows:

> ‘‘A consistent user interface will ensure that all user-accessible functions and services will appear and behave in a similar, predictable fashion regardless of application or site. This will lead to better efficiency and fewer user errors, which in turn may result in lower recovery costs.’’

To make this objective SMART, we ask whether the objective is specific, measurable, actionable, realistic, and time-bound, and then augment the objective appropriately.

The following captures an analysis of these criteria for the stated objective:

- **Specific**: The objective of providing ‘‘a consistent user interface that will ensure all user accessible functions and services will appear and behave in a similar, predictable fashion regardless of application or site’’. is pretty specific. However, the measures listed in the second sentence could be more specific . . .

- **Measurable**: As stated above, the objective is measurable, but could be more specific. The second sentence could be amended to read (for example): ‘‘This will lead to 10% greater user efficiency and 20% fewer order entry user errors, which in turn may result in 5% lower order entry costs’’.
- **Actionable**: The objective does appear to be actionable. It seems clear that consistency of the user interface must be provided, and that could be handled by whoever is responsible for providing the user interface to the user device.
- **Realistic**: The objective of providing ‘‘a consistent user interface that will ensure all user accessible functions and services will appear and behave in a similar, predictable fashion regardless of application or site’’ might not be realistic. Considering the use today of PDAs at the user end might lead us to augment this objective to ensure that the downstream developers don’t unduly create designs that hinder the use of new technologies. The objective could be re-stated as ‘‘a consistent user interface, across user interface devices that provide similar functionality, that will ensure . . .’’ etc.
- **Time-bound**: The objective as stated is not time-bound. To be time-bound the objective could be re-stated as ‘‘By the end of Q3, provide a consistent . . .’’.

The above results in a SMART objective that looks more like this (again remember this is an example):

> ‘‘By the end of Q3, provide a consistent user interface across user interface devices that provide similar functionality to ensure all user accessible functions and services appear and behave in a similar way when using those devices in a predictable fashion regardless of application or site. This will lead to 10% greater user efficiency and 20% fewer order entry user errors, which in turn may result in 5% lower order entry costs.’’

5.3.3 Categories of Goals and Objectives

Although every organization will have its own set of goals, some examples may help in the development of an organization-specific list. The goals given below are categories of goals, each with a list of possible objectives, which have been adapted from the goals given in previous versions of TOGAF.

Each of the objectives given below should be made SMART with specific measures and metrics for the task involved, as illustrated in the example above. However, the actual work to be done will be specific to the architecture project concerned, and it is not possible to provide a list of generic SMART objectives that will relate to any project.

Goal: Improve Business Process Performance

Business process improvements can be realized through the following objectives:

- Increased process throughput
- Consistent output quality
- Predictable process costs
- Increased re-use of existing processes
- Reduced time of sending business information from one process to another process

Goal: Decrease Costs

Cost improvements can be realized through the following objectives:

- Lower levels of redundancy and duplication in assets throughout the enterprise
- Decreased reliance on external IT service providers for integration and customization
- Lower costs of maintenance

Goal: Improve Business Operations

Business operations improvements can be realized through the following objectives:

- Increased budget available to new business features
- Decreased costs of running the business
- Decreased time-to-market for products or services
- Increased quality of services to customers
- Improved quality of business information

Goal: Improve Management Efficacy

Management efficacy improvements can be realized through the following objectives:

- Increased flexibility of business
- Shorter time to make decisions
- Higher quality decisions

Goal: Reduce Risk

Risk improvements can be realized through the following objectives:

- Ease of implementing new processes
- Decreased errors introduced into business processes through complex and faulty systems
- Decreased real-world safety hazards (including hazards that cause loss of life)

Goal: Improve Effectiveness of IT Organization

IT organization effectiveness can be realized through the following objectives:

- Increased rollout of new projects
- Decreased time to rollout new projects
- Lower cost in rolling out new projects
- Decreased loss of service continuity when rolling out new projects
- Common development: applications that are common to multiple business areas will be developed or acquired once and re-used rather than separately developed by each business area.
- Open systems environment: a standards-based common operating environment, which accommodates the injection of new standards, technologies, and applications on an organization-wide basis, will be established. This standards-based environment will provide the basis for development of common applications and facilitate software re-use.

- Use of products: as far as possible, hardware-independent, off-the-shelf items should be used to satisfy requirements in order to reduce dependence on custom developments and to reduce development and maintenance costs.
- Software re-use: for those applications that must be custom developed, development of portable applications will reduce the amount of software developed and add to the inventory of software suitable for re-use by other systems.
- Resource sharing: data processing resources (hardware, software, and data) will be shared by all users requiring the services of those resources. Resource sharing will be accomplished in the context of security and operational considerations.

Goal: Improve User Productivity

User productivity improvements can be realized through the following objectives:

- Consistent user interface: a consistent user interface will ensure that all user-accessible functions and services will appear and behave in a similar, predictable fashion regardless of application or site. This will lead to better efficiency and fewer user errors, which in turn may result in lower recovery costs.
- Integrated applications: applications available to the user will behave in a logically consistent manner across user environments, which will lead to the same benefits as a consistent user interface.
- Data sharing: databases will be shared across the organization in the context of security and operational considerations, leading to increased ease-of-access to required data.

Goal: Improve Portability and Scalability

The portability and scalability of applications will be through the following objectives:

- Portability: applications that adhere to open systems standards will be portable, leading to increased ease-of-movement across heterogeneous computing platforms. Portable applications can allow sites to upgrade their platforms as technological improvements occur, with minimal impact on operations.
- Scalability: applications that conform to the model will be configurable, allowing operation on the full spectrum of platforms required.

Goal: Improve Interoperability

Interoperability improvements across applications and business areas can be realized through the following objectives:

- Common infrastructure: the architecture should promote a communications and computing infrastructure based on open systems and systems transparency including, but not limited to, operating systems, database management, data interchange, network services, network management, and user interfaces.
- Standardization: by implementing standards-based platforms, applications will be provided with and will be able to use a common set of services that improve the opportunities for interoperability.

Goal: Increase Vendor Independence

Vendor independence will be increased through the following objectives:

- Interchangeable components: only hardware and software that have standards-based interfaces will be selected, so that upgrades or the insertion of new products will result in minimal disruption to the user's environment.
- Non-proprietary specifications: capabilities will be defined in terms of non-proprietary specifications that support full and open competition and are available to any vendor for use in developing commercial products.

Goal: Reduce Lifecycle Costs

Lifecycle costs can be reduced through most of the objectives discussed above. In addition, the following objectives directly address reduction of lifecycle costs:

- Reduced duplication: replacement of isolated systems and islands of automation with interconnected open systems will lead to reductions in overlapping functionality, data duplication, and unneeded redundancy because open systems can share data and other resources.
- Reduced software maintenance costs: reductions in the quantity and variety of software used in the organization will lead to reductions in the amount and cost of software maintenance. Use of standard off-the-shelf software will lead to further reductions in costs since vendors of such software distribute their product maintenance costs across a much larger user base.
- Incremental replacement: common interfaces to shared infrastructure components allow for phased replacement or upgrade with minimal operational disturbance.
- Reduced training costs: common systems and consistent Human Computer Interfaces (HCIs) will lead to reduced training costs.

Goal: Improve Security

Security can be improved in the organization's information through the following objectives:

- Consistent security interfaces for applications: consistent security interfaces and procedures will lead to fewer errors when developing applications and increased application portability. Not all applications will need the same suite of security features, but any features used will be consistent across applications.
- Consistent security interfaces for users: a common user interface to security features will lead to reduced learning time when moving from system to system.
- Security independence: application deployment can use the security policy and mechanisms appropriate to the particular environment if there is good layering in the architecture.
- A 25% reduction in calls to the help desk relating to security issues.
- A 20% reduction in ''false positives'' detected in the network (a false positive is an event that appears to be an actionable security event, but in fact is a false alarm).

Goal: Improve Manageability

Management improvement can be realized through the following objectives:

- Consistent management interface: consistent management practices and procedures will facilitate management across all applications and their underlying support structures. A consistent interface can simplify the management burden, leading to increased user efficiency.
- Reduced operation, administration, and maintenance costs: operation, administration, and maintenance costs may be reduced through the availability of improved management products and increased standardization of the objects being managed.

Chapter 6

Architecture Deliverables

This amended extract from TOGAF 9.2 provides descriptions of architecture deliverables for Level 1 Syllabus, section 3.1.11, learning unit 11. This handout is intended to be used for TOGAF 9.2 Level 1 training. It can also be used for TOGAF 9.2 Level 2 training.

This handout describes the purpose of deliverables that will typically be consumed and produced across the TOGAF ADM cycle. As deliverables are typically the contractual or formal work products of an architecture project, it is likely that these deliverables will be constrained or altered by any overarching project or process management for the enterprise (such as CMMI, PRINCE2, PMBOK, or MSP).

The deliverables described are intended to provide a typical baseline of architecture deliverables in order to better define the activities required in the ADM and act as a starting point for tailoring within a specific organization.

6.1 Deliverables

The following sections provide example descriptions of deliverables referenced in the ADM.

6.1.1 Architecture Building Blocks

Architecture documentation and models from the enterprise's Architecture Repository.

6.1.2 Architecture Contract

Architecture Contracts are the joint agreements between development partners and sponsors on the deliverables, quality, and fitness-for-purpose of an architecture. Successful implementation of these agreements will be delivered through effective architecture governance. By implementing a governed approach to the management of contracts, the following will be ensured:

- A system of continuous monitoring to check integrity, changes, decision-making, and audit of all architecture-related activities within the organization
- Adherence to the principles, standards, and requirements of the existing or developing architectures
- Identification of risks in all aspects of the development and implementation of the architecture(s) covering the internal development against accepted standards, policies, technologies, and products as well as the operational aspects of the architectures such that the organization can continue its business within a resilient environment

- A set of processes and practices that ensure accountability, responsibility, and discipline with regard to the development and usage of all architectural artifacts
- A formal understanding of the governance organization responsible for the contract, their level of authority, and scope of the architecture under the governance of this body

6.1.3 Architecture Definition Document

The Architecture Definition Document is the deliverable container for the core architectural artifacts created during a project and for important related information. The Architecture Definition Document spans all architecture domains (business, data, application, and technology) and also examines all relevant states of the architecture (baseline, transition, and target).

A Transition Architecture shows the enterprise at an architecturally significant state between the Baseline and Target Architectures. Transition Architectures are used to describe transitional Target Architectures necessary for effective realization of the Target Architecture.

The Architecture Definition Document is a companion to the Architecture Requirements Specification, with a complementary objective:

- The Architecture Definition Document provides a qualitative view of the solution and aims to communicate the intent of the architects.
- The Architecture Requirements Specification provides a quantitative view of the solution, stating measurable criteria that must be met during the implementation of the architecture.

6.1.4 Architecture Principles

Principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission.

In their turn, principles may be just one element in a structured set of ideas that collectively define and guide the organization, from values through to actions and results.

6.1.5 Architecture Repository

The Architecture Repository acts as a holding area for all architecture-related projects within the enterprise. The repository allows projects to manage their deliverables, locate re-usable assets, and publish outputs to stakeholders and other interested parties.

6.1.6 Architecture Requirements Specification

The Architecture Requirements Specification provides a set of quantitative statements that outline what an implementation project must do in order to comply with the architecture. An Architecture Requirements Specification will typically form a major component of an implementation contract or contract for more detailed Architecture Definition.

As mentioned above, the Architecture Requirements Specification is a companion to the Architecture Definition Document, with a complementary objective:

- The Architecture Definition Document provides a qualitative view of the solution and aims to communicate the intent of the architect.

- The Architecture Requirements Specification provides a quantitative view of the solution, stating measurable criteria that must be met during the implementation of the architecture.

6.1.7 Architecture Roadmap

The Architecture Roadmap lists individual work packages that will realize the Target Architecture and lays them out on a timeline to show progression from the Baseline Architecture to the Target Architecture. The Architecture Roadmap highlights individual work packages' business value at each stage. Transition Architectures necessary to effectively realize the Target Architecture are identified as intermediate steps. The Architecture Roadmap is incrementally developed throughout Phases E and F, and informed by readily identifiable roadmap components from Phase B, C, and D within the ADM.

6.1.8 Architecture Vision

The Architecture Vision is created early on in the ADM cycle. It provides a summary of the changes to the enterprise that will accrue from successful deployment of the Target Architecture. The purpose of the Architecture Vision is to provide key stakeholders with a formally agreed outcome. Early agreement on the outcome enables the architects to focus on the detail necessary to validate feasibility. Providing an Architecture Vision also supports stakeholder communication by providing a summary version of the full Architecture Definition.

6.1.9 Business Principles, Business Goals, and Business Drivers

Business principles, business goals, and business drivers provide context for architecture work, by describing the needs and ways of working employed by the enterprise. Many factors that lie outside the consideration of architecture discipline may nevertheless have significant implications for the way that architecture is developed.

6.1.10 Capability Assessment

Before embarking upon a detailed Architecture Definition, it is valuable to understand the baseline and target capability level of the enterprise. This Capability Assessment can be examined on several levels:

- What is the capability level of the enterprise as a whole? Where does the enterprise wish to increase or optimize capability? What are the architectural focus areas that will support the desired development of the enterprise?
- What is the capability or maturity level of the IT function within the enterprise? What are the likely implications of conducting the architecture project in terms or design governance, operational governance, skills, and organization structure? What is an appropriate style, level of formality, and amount of detail for the architecture project to fit with the culture and capability of the IT organization?
- What is the capability and maturity of the architecture function within the enterprise? What architectural assets are currently in existence? Are they maintained and accurate? What standards and reference models need to be considered? Are there likely to be opportunities to create re-usable assets during the architecture project?

- Where capability gaps exist, to what extent is the business ready to transform in order to reach the target capability? What are the risks to transformation, cultural barriers, and other considerations to be addressed beyond the basic capability gap?

6.1.11 Change Request

During implementation of an architecture, as more facts become known, it is possible that the original Architecture Definition and requirements are not suitable or are not sufficient to complete the implementation of a solution. In these circumstances, it is necessary for implementation projects to either deviate from the suggested architectural approach or to request scope extensions. Additionally, external factors — such as market factors, changes in business strategy, and new technology opportunities — may open up opportunities to extend and refine the architecture.

In these circumstances, a Change Request may be submitted in order to kick-start a further cycle of architecture work.

6.1.12 Communications Plan

Enterprise architectures contain large volumes of complex and inter-dependent information. Effective communication of targeted information to the right stakeholders at the right time is a critical success factor for enterprise architecture. Development of a Communications Plan for architecture allows for this communication to be carried out within a planned and managed process.

6.1.13 Compliance Assessment

Once an architecture has been defined, it is necessary to govern that architecture through implementation to ensure that the original Architecture Vision is appropriately realized and that any implementation learnings are fed back into the architecture process. Period compliance reviews of implementation projects provide a mechanism to review project progress and ensure that the design and implementation is proceeding in-line with the strategic and architectural objectives.

6.1.14 Implementation and Migration Plan

The Implementation and Migration Plan provides a schedule of the projects that will realize the Target Architecture. The Implementation and Migration Plan includes executable projects grouped into managed portfolios and programs. The Implementation and Migration Strategy

identifying the approach to change is a key element of the Implementation and Migration Plan.

6.1.15 Implementation Governance Model

Once an architecture has been defined, it is necessary to plan how the Transition Architecture that implements the architecture will be governed through implementation. Within organizations that have established architecture functions, there is likely to be a governance framework already in place, but specific processes, organizations, roles, responsibilities, and measures may need to be defined on a project-by-project basis.

The Implementation Governance Model ensures that a project transitioning into implementation also smoothly transitions into appropriate architecture governance.

6.1.16 Organizational Model for Enterprise Architecture

In order for an architecture framework to be used successfully, it must be supported by the correct organization, roles, and responsibilities within the enterprise. Of particular importance is the definition of boundaries between different enterprise architecture practitioners and the governance relationships that span across these boundaries.

6.1.17 Request for Architecture Work

This is a document that is sent from the sponsoring organization to the architecture organization to trigger the start of an architecture development cycle. Requests for Architecture Work can be created as an output of the Preliminary Phase, a result of approved architecture Change Requests, or terms of reference for architecture work originating from migration planning.

In general, all the information in this document should be at a high level.

6.1.18 Requirements Impact Assessment

Throughout the ADM, new information is collected relating to an architecture. As this information is gathered, new facts may come to light that invalidate existing aspects of the architecture. A Requirements Impact Assessment assesses the current architecture requirements and specification to identify changes that should be made and the implications of those changes.

6.1.19 Solution Building Blocks

Implementation-specific building blocks from the enterprise's Architecture Repository.

6.1.20 Statement of Architecture Work

The Statement of Architecture Work defines the scope and approach that will be used to complete an architecture development cycle. The Statement of Architecture Work is typically the document against which successful execution of the architecture project will be measured and may form the basis for a contractual agreement between the supplier and consumer of architecture services.

6.1.21 Tailored Architecture Framework

TOGAF provides an industry standard framework for architecture that may be used in a wide variety of organizations. However, before TOGAF can be effectively used within an architecture project, tailoring at two levels is necessary.

Firstly, it is necessary to tailor the TOGAF model for integration into the enterprise. This tailoring will include integration with project and process management frameworks, customization of terminology, development of presentational styles, selection, configuration, and deployment of architecture tools, etc. The formality and detail of any frameworks adopted should also align with other contextual factors for the enterprise, such as culture, stakeholders, commercial models for enterprise architecture, and the existing level of Architecture Capability.

Once the framework has been tailored to the enterprise, further tailoring is necessary in order to tailor the framework for the specific architecture project. Tailoring at this level will select appropriate deliverables and artifacts to meet project and stakeholder needs.

Chapter 7

Architecture Skills Framework

This handout is extracted from chapter 52 of TOGAF 9.2 and provides a set of role, skill, and experience norms for staff undertaking enterprise architecture work. This handout is used for TOGAF 9.2 Level 2 training.

7.1 Introduction

Skills frameworks provide a view of the competency levels required for specific roles. They define:

- The roles within a work area
- The skills required by each role
- The depth of knowledge required to fulfil the role successfully

They are relatively common for defining the skills required for a consultancy and/or project management assignment, to deliver a specific project or work package. They are also widely used by recruitment and search agencies to match candidates and roles.

Their value derives from their ability to provide a means of rapidly identifying skill matches and gaps. Successfully applied, they can ensure that candidates are fit for the jobs assigned to them.

Their value in the context of enterprise architecture arises from the immaturity of the enterprise architecture discipline, and the problems that arise from this.

7.2 Need for an Enterprise Architecture Skills Framework

7.2.1 Definitional Rigor

‘‘Enterprise Architecture’’ and ‘‘Enterprise Architect’’ are widely used but poorly defined terms in industry today. They are used to denote a variety of practices and skills applied in a wide variety of architecture domains. There is a need for better classification to enable more implicit understanding of what type of architecture/architect is being described.

This lack of uniformity leads to difficulties for organizations seeking to recruit or assign/promote staff to fill positions in the architecture field. Because of the different usages of terms, there is often misunderstanding and miscommunication between those seeking to recruit for, and those seeking to fill, the various roles of the architect.

7.2.2 Basis of an Internal Architecture Practice

Despite the lack of uniform terminology, architecture skills are in increasing demand, as the discipline of architecture gains increasing attention within industry.

Many enterprises have set up, or are considering setting up, an enterprise architecture practice, as a means of fostering development of the necessary skills and experience among in-house staff to undertake the various architecting tasks required by the enterprise.

An enterprise architecture practice is a formal program of development and certification, by which an enterprise formally recognizes the skills of its practicing architects, as demonstrated by their work. Such a program is essential in order to ensure the alignment of staff skills and experience with the architecture tasks that the enterprise wishes to be performed.

The role and skill definitions on which such a program needs to be based are also required, by both recruiting and supplying organizations, in cases where external personnel are to be engaged to perform architecture work (for example, as part of a consultancy engagement).

An enterprise architecture practice is both difficult and costly to set up. It is normally built around a process of peer review, and involves the time and talent of the strategic technical leadership of an enterprise. Typically it involves establishment of a peer review board, and documentation of the process, and of the requirements for internal certification. Time is also required of candidates to prepare for peer review, by creating a portfolio of their work to demonstrate their skills, experiences, and contributions to the profession.

The TOGAF Architecture Skills Framework attempts to address this need by providing definitions of the architecting skills and proficiency levels required of personnel, internal or external, who are to perform the various architecting roles defined within the TOGAF Framework.

Because of the complexity, time, and cost involved, many enterprises do not have an internal enterprise architect certification program, preferring instead to simply interview and recruit architecture staff on an *ad hoc* basis. There are serious risks associated with this approach:

- Communication between recruiting organizations, consultancies, and employment agencies is very difficult.
- Time is wasted interviewing staff who may have applied in all good faith, but still lack the skills and/or experience required by the employer.
- Staff that are capable of filling architecture roles may be overlooked, or may not identify themselves with advertised positions and hence not even apply.
- There is increased risk of unsuitable personnel being employed or engaged, through no-one's fault, and despite everyone involved acting in good faith. This in turn can:
  - Increase personnel costs, through the need to rehire or reassign staff
  - Adversely impact the time, cost, and quality of operational IT systems, and the projects that deliver them

7.3 Goals/Rationale

7.3.1 Certification of Enterprise Architects

The main purpose behind an enterprise setting up an internal enterprise architect certification program is two-fold:

1. To formally recognize the skill of its practicing architects, as part of the task of establishing and maintaining a professional architecting organization
2. To ensure the alignment of necessary staff skills and experience with the architecture tasks that the enterprise wishes to be performed, whether these are to be performed internally to the enterprise or externally; for example, as part of a consultancy engagement

7.3.2 Specific Benefits

Specific benefits anticipated from use of the TOGAF Architecture Skills Framework include:

- Reduced time, cost, and risk in training, hiring, and managing architecture professionals, both internal and external:
  - Simplifies communication between recruiting organizations, consultancies, and employment agencies
  - Avoids wasting time interviewing staff who may have applied in all good faith, but still lack the skills and/or experience required by the employer
  - Avoids staff who are capable of filling architecture roles being overlooked, or not identifying themselves with advertised positions and hence not even applying
- Reduced time and cost to set up an internal architecture practice:
  - Many enterprises do not have an internal architecture practice due to the complexity involved in setting one up, preferring instead to simply interview and recruit architecture staff on an *ad hoc* basis.
  - By providing definitions of the architecting skills and proficiency levels required of personnel who are to perform the various architecting roles defined within TOGAF, the Architecture Skills Framework greatly reduces the time, cost, and risk of setting up a practice for the first time, and avoids ‘‘re-inventing wheels’’.
  - Enterprises that already have an internal architecture practice are able to set enterprise-wide norms, but still experience difficulties as outlined above in recruiting staff, or engaging consultants, from external sources, due to the lack of uniformity between different enterprises. By aligning its existing skills framework with the industry-accepted definitions provided by The Open Group, an enterprise can greatly simplify these problems.
- Reduced time and cost to implement an architecture practice helps reduce the time, cost, and risk of overall solution development:
  - Enterprises that do not have an internal architecture practice run the risk of unsuitable personnel being employed or engaged, through no-one’s fault, and despite everyone involved acting in good faith. The resultant time and cost penalties far outweigh the time and cost of having an internal architecture practice:

— Personnel costs are increased, through the occasional need to rehire or reassign staff.

— Even more important is the adverse impact on the time, cost, and quality of operational IT systems, and the projects to deliver them, resulting from poor staff assignments.

7.4 Enterprise Architecture Role and Skill Categories

7.4.1 Overview

This section describes the role of an enterprise architect, the fundamental skills required, and some possible disciplines in which an enterprise architect might specialize.

TOGAF delivers an enterprise architecture, and therefore requires both business and IT-trained professionals to develop the enterprise architecture.

The TOGAF Architecture Skills Framework provides a view of the competency levels for specific roles within the enterprise architecture team. The Framework defines:

- The roles within an enterprise architecture work area
- The skills required by those roles
- The depth of knowledge required to fulfil each role successfully

The value is in providing a rapid means of identifying skills and gaps. Successfully applied, the Framework can be used as a measure for:

- Staff development
- Ensuring that the right person does the right job

7.4.2 TOGAF Roles

A typical architecture team undertaking the development of an enterprise architecture as described in TOGAF would comprise the following roles:

- Architecture Board Members
- Architecture Sponsor
- Architecture Manager
- Architects for:
  - — Enterprise Architecture (which for the purpose of the tables shown below can be considered as a superset of Business, Data, Application, and Technology Architecture)
  - — Business Architecture
  - — Data Architecture

  - — Application Architecture
  - — Technology Architecture
- Program and/or Project Managers
- IT Designer
- And many others . . .

The tables that follow show, for each of these roles, the skills required and the desirable level of proficiency in each skill.

Of all the roles listed above, the one that needs particularly detailed analysis and definition is of course the central role of enterprise architect. As explained above, ''Enterprise Architecture'' and ''Enterprise Architect'' are terms that are very widely used but very poorly defined in industry today, denoting a wide variety of practices and skills applied in a wide variety of architecture domains. There is often confusion between the role of an architect and that of a designer or builder. Many of the skills required by an enterprise architect are also required by the designer, who delivers the solutions. While their skills are complementary, those of the designer are primarily technology focused and translate the architecture into deliverable components.

The final subsection below therefore explores in some detail the generic characteristics of the role of enterprise architect, and the key skill requirements, whatever the particular architecture domain (Enterprise Architecture, Business Architecture, Data Architecture, Application Architecture, Technology Architecture, etc.).

7.4.3 Categories of Skills

The TOGAF team skill set will need to include the following main categories of skills:

- **Generic Skills**: — typically comprising leadership, teamworking, inter-personal skills, etc.
- **Business Skills & Methods**: — typically comprising business cases, business process, strategic planning, etc.
- **Enterprise Architecture Skills**: — typically comprising modeling, building block design, applications and role design, systems integration, etc.
- **Program or Project Management Skills**: — typically comprising managing business change, project management methods and tools, etc.
- **IT General Knowledge Skills**: — typically comprising brokering applications, asset management, migration planning, SLAs, etc.
- **Technical IT Skills**: — typically comprising software engineering, security, data interchange, data management, etc.
- **Legal Environment**: — typically comprising data protection laws, contract law, procurement law, fraud, etc.

The tables that follow illustrate each of these categories of skills.

The tables that follow show, for each of these skills, the roles to which they are relevant and the desirable level of proficiency in each skill.

7.4.4 Proficiency Levels

The TOGAF Architecture Skills Framework identifies four levels of knowledge or proficiency in any area:

Level	Achievement	Description
	Background	Not a required skill, though should be able to define and manage skill if required.
	Awareness	Understands the background, issues, and implications sufficiently to be able to understand how to proceed further and advise client accordingly.
3	Knowledge	Detailed knowledge of subject area and capable of providing professional advice and guidance. Ability to integrate capability into architecture design.
4	Expert	Extensive and substantial practical experience and applied knowledge on the subject.

7.5 Enterprise Architecture Role and Skill Definitions

7.5.1 Generic Skills

Roles	Architecture Board Member	Architecture Sponsor	Enterprise Architecture Manager	Enterprise Architecture Technology	Enterprise Architecture Data	Enterprise Architecture Applications	Enterprise Architecture Business	Program/ Project Manager	IT Designer
Generic Skills									
Leadership	4	4	4	3	3	3	3	4	1
Teamwork	3	3	4	4	4	4	4	4	2
Inter-personal	4	4	4	4	4	4	4	4	2
Oral Communications	3	3	4	4	4	4	4	4	2
Written Communications	3	3	4	4	4	4	4	3	3
Logical Analysis	2	2	4	4	4	4	4	3	3
Stakeholder Management	4	3	4	3	3	3	3	4	2
Risk Management	3	3	4	3	3	3	3	4	1

7.5.2 Business Skills & Methods

Roles	Architecture Board Member	Architecture Sponsor	Enterprise Architecture Manager	Enterprise Architecture Technology	Enterprise Architecture Data	Enterprise Architecture Applications	Enterprise Architecture Business	Program/ Project Manager	IT Designer
Business Skills & Methods									
Business Case	3	4	4	4	4	4	4	4	2
Business Scenario	2	3	4	4	4	4	4	3	2
Organization	3	3	4	3	3	3	4	3	2
Business Process	3	3	4	4	4	4	4	3	2
Strategic Planning	2	3	3	3	3	3	4	3	1
Budget Management	3	3	3	3	3	3	3	4	3
Visioning	3	3	4	3	3	3	4	3	2
Business Metrics	3	4	4	4	4	4	4	4	3
Business Culture	4	4	4	3	3	3	3	3	1
Legacy Investments	4	4	3	2	2	2	2	3	2
Business Functions	3	3	3	3	4	4	4	3	2

7.5.3 Enterprise Architecture Skills

Roles	Architecture Board Member	Architecture Sponsor	Enterprise Architecture Manager	Enterprise Architecture Technology	Enterprise Architecture Data	Enterprise Architecture Applications	Enterprise Architecture Business	Program/ Project Manager	IT Designer
Enterprise Architecture Skills									
Business Modeling	2	2	4	3	3	4	4	2	2
Business Process Design	1	1	4	3	3	4	4	2	2
Role Design	2	2	4	3	3	4	4	2	2
Organization Design	2	2	4	3	3	4	4	2	2
Data Design	1	1	3	3	4	3	3	2	3
Application Design	1	1	3	3	3	4	3	2	3
Systems Integration	1	1	4	4	3	3	3	2	2
IT Industry Standards	1	1	4	4	4	4	3	2	3
Services Design	2	2	4	4	3	4	3	2	2
Architecture Principles Design	2	2	4	4	4	4	4	2	2
Architecture Views & Viewpoints Design	2	2	4	4	4	4	4	2	2
Building Block Design	1	1	4	4	4	4	4	2	3
Solutions Modeling	1	1	4	4	4	4	4	2	3
Benefits Analysis	2	2	4	4	4	4	4	4	2
Business Interworking	3	3	4	3	3	4	4	3	1
Systems Behavior	1	1	4	4	4	4	3	3	2
Project Management	1	1	3	3	3	3	3	4	2

7.5.4 Program or Project Management Skills

Roles	Architecture Board Member	Architecture Sponsor	Enterprise Architecture Manager	Enterprise Architecture Technology	Enterprise Architecture Data	Enterprise Architecture Applications	Enterprise Architecture Business	Program/ Project Manager	IT Designer
Program or Project Management Skills									
Program Management	1	2	3	3	3	3	3	4	2
Project Management	1	2	3	3	3	3	3	4	2
Managing Business Change	3	3	4	3	3	3	4	4	2
Change Management	3	3	4	3	3	3	4	3	2
Value Management	4	4	4	3	3	3	4	3	2

7.5.5 IT General Knowledge Skills

Roles	Architecture Board Member	Architecture Sponsor	Enterprise Architecture Manager	Enterprise Architecture Technology	Enterprise Architecture Data	Enterprise Architecture Applications	Enterprise Architecture Business	Program/ Project Manager	IT Designer
IT General Knowledge Skills									
IT Application Development Methodologies & Tools	2	2	3	4	4	4	2	3	3
Programming Languages	1	1	3	4	4	4	3	2	3
Brokering Applications	1	1	3	3	4	4	3	2	3
Information Consumer Applications	1	1	3	3	4	4	3	2	3
Information Provider Applications	1	1	3	3	4	4	3	2	3
Storage Management	1	1	3	4	4	2	2	2	3
Networks	1	1	3	4	3	2	2	2	3
Web-based Services	1	1	3	3	4	4	2	2	3
IT Infrastructure	1	1	3	4	3	2	2	2	3
Asset Management	1	1	4	4	3	3	3	2	3
Service Level Agreements	1	1	4	4	3	4	3	2	3
Systems	1	1	3	4	3	3	2	2	3
COTS	1	1	3	4	3	4	2	2	3
Enterprise Continuums	1	1	4	4	4	4	4	2	3
Migration Planning	1	1	4	3	4	3	3	2	3
Management Utilities	1	1	3	2	4	4	2	2	3
Infrastructure	1	1	3	4	3	4	2	2	3

7.5.6 Technical IT Skills

Roles	Architecture Board Member	Architecture Sponsor	Enterprise Architecture Manager	Enterprise Architecture Technology	Enterprise Architecture Data	Enterprise Architecture Applications	Enterprise Architecture Business	Program/ Project Manager	IT Designer
Technical IT Skills									
Software Engineering	1	1	3	3	4	4	3	2	3
Security	1	1	3	4	3	4	3	2	3
Systems & Network Management	1	1	3	4	3	3	3	2	3
Transaction Processing	1	1	3	4	3	4	3	2	3
Location & Directory	1	1	3	4	4	3	3	2	3
User Interface	1	1	3	4	4	4	3	2	3
International Operations	1	1	3	4	3	3	2	2	2
Data Interchange	1	1	3	4	4	3	2	2	3
Data Management	1	1	3	4	4	3	2	2	3
Graphics & Image	1	1	3	4	3	3	2	2	3
Operating System Services	1	1	3	4	3	3	2	2	3
Network Services	1	1	3	4	3	3	2	2	3
Communications Infrastructure	1	1	3	4	3	3	2	2	3

7.5.7 Legal Environment

Roles	Architecture Board Member	Architecture Sponsor	Enterprise Architecture Manager	Enterprise Architecture Technology	Enterprise Architecture Data	Enterprise Architecture Applications	Enterprise Architecture Business	Program/ Project Manager	IT Designer
Legal Environment									
Contract Law	2	2	2	2	2	2	2	3	1
Data Protection Law	3	3	4	3	3	3	3	2	2
Procurement Law	3	2	2	2	2	2	2	4	1
Fraud	3	3	3	3	3	3	3	3	1
Commercial Law	3	3	2	2	2	2	3	3	1

7.6 Generic Role and Skills of the Enterprise Architect

Of all the roles listed above, the one that needs particularly detailed analysis and definition is, of course, the central role of enterprise architect. As explained above, ''Enterprise Architecture'' and ''Enterprise Architect'' are terms that are very widely used but very poorly defined in industry today, denoting a wide variety of practices and skills applied in a wide variety of architecture domains.

This section therefore explores in some detail the generic characteristics of the role of enterprise architect, and some key skill requirements, whatever the particular architecture domain (Enterprise Architecture, Business Architecture, Data Architecture, Application Architecture, Technology Architecture, etc.).

7.6.1 Generic Role

Enterprise architects are visionaries, coaches, team leaders, business-to-technical liaisons, computer scientists, and industry experts.

The following is effectively a job description for an enterprise architect:

> ''The architect has a responsibility for ensuring the completeness (fitness-for-purpose) of the architecture, in terms of adequately addressing all the pertinent concerns of its stakeholders; and the integrity of the architecture, in terms of connecting all the various views to each other, satisfactorily reconciling the conflicting concerns of different stakeholders, and showing the trade-offs made in so doing (as between security and performance, for example).
>
> The choice of which particular architecture views to develop is one of the key decisions that the enterprise architect has to make. The choice has to be constrained by considerations of practicality, and by the principle of fitness-for-purpose (i.e., the architecture should be developed only to the point at which it is fit-for-purpose, and not reiterated *ad infinitum* as an academic exercise).''

The role of the enterprise architect is more like that of a city planner than that of a building architect, and the product of the enterprise architect is more aptly characterized as a planned community (as opposed to an unconstrained urban sprawl), rather than as a well-designed building or set of buildings.

An enterprise architect does not create the technical vision of the enterprise, but has professional relationships with executives of the enterprise to gather and articulate the technical vision, and to produce the strategic plan for realizing it. This plan is always tied to the business plans of the enterprise, and design decisions are traceable to the business plan.

The strategic plan of the enterprise architect is tied to the architecture governance process for the enterprise, so design decisions are not circumvented for tactical convenience.

The enterprise architect produces documentation of design decisions for application development teams or product implementation teams to execute.

An architect is involved in the entire process; beginning with working with the customer to understand real needs, as opposed to wants, and then throughout the process to translate those needs into capabilities verified to meet the needs. Additionally, the architect may present different models to the customer that communicate how those needs may be met, and is therefore an essential participant in the consultative selling process.

However, the architect is not the builder, and must remain at a level of abstraction necessary to ensure that they do not get in the way of practical implementation.

The following excerpt from *The Art of Systems Architecting* depicts this notion:

> ‘‘It is the responsibility of the architect to know and concentrate on the critical few details and interfaces that really matter, and not to become overloaded with the rest.’’

The architect’s focus is on understanding what it takes to satisfy the client, where qualitative worth is used more than quantitative measures. The architect uses more inductive skills than the deductive skills of the builder. The architect deals more with guidelines, rather than rules that builders use as a necessity.

It also must be clear that the role of an architect may be performed by an engineer. A goal of this document is to describe the role — what should be done, regardless of who is performing it.

Thus, the role of the architect can be summarized as to:

- **Understand and interpret requirements**: probe for information, listen to information, influence people, facilitate consensus building, synthesize and translate ideas into actionable requirements, articulate those ideas to others. Identify use or purpose, constraints, risks, etc. The architect participates in the discovery and documentation of the customer’s business scenarios that are driving the solution. The architect is responsible for requirements understanding and embodies that requirements understanding in the architecture specification.
- **Create a useful model**: take the requirements and develop well-formulated models of the components of the solution, augmenting the models as necessary to fit all of the circumstances. Show multiple views through models to communicate the ideas effectively. The architect is responsible for the overall architecture integrity and maintaining the vision of the offering from an architectural perspective. The architect also ensures leverage opportunities are identified, using building blocks, and is a liaison between the functional groups (especially development and marketing) to ensure that the leverage opportunities are realized. The architect provides and maintains these models as a framework for understanding the domain(s) of development work, guiding what should be done within the organization, or outside the organization. The architect must represent the organization view of the architecture by understanding all the necessary business components.
- **Validate, refine, and expand the model**: verify assumptions, bring in subject matter experts, etc. in order to improve the model and to further define it, adding as necessary new ideas to make the result more flexible and more tightly linked to current and expected requirements. The architect additionally should assess the value of solution-enhancing developments emanating from field work and incorporate these into the architecture models as appropriate.
- **Manage the architecture**: continuously monitor the models and update them as necessary to show changes, additions, and alterations. Represent architecture and issues during development and decision points of the program. The architect is an ‘‘agent of change’’, representing that need for the implementation of the architecture. Through this development cycle, the architect continuously fosters the sharing of customer, architecture, and technical information between organizations.

7.6.2 Characterization in Terms of the Enterprise Continuum

Under certain circumstances, the complexity of a solution may require additional architects to support the architecture effort. The different categories of architects are described below, but as they are architects, they all perform the tasks described above. Any combination of enterprise, enterprise solution, and solution architects may be utilized, as a team. In such cases each member may have a specific focus, if not specific roles and responsibilities, within the phases of the development process. In cases where a team of architects is deemed necessary, a lead enterprise architect should be assigned to manage and lead the team members.

- The **Enterprise Architect** has the responsibility for architectural design and documentation at a landscape and technical reference model level. The Enterprise Architect often leads a group of the Segment Architects and/or Solution Architects related to a given program. The focus of the Enterprise Architect is on enterprise-level business functions required.
- The **Segment Architect** has the responsibility for architectural design and documentation of specific business problems or organizations. A Segment Architect re-uses the output from all other architects, joining detailed technical solutions to the overall architectural landscape. The focus of the Segment Architect is on enterprise-level business solutions in a given domain, such as finance, human resources, sales, etc.
- The **Solution Architect** has the responsibility for architectural design and documentation at a system or subsystem level, such as management or security. A Solution Architect may shield the Enterprise/Segment Architect from the unnecessary details of the systems, products, and/or technologies. The focus of the Solution Architect is on system technology solutions; for example, a component of a solution such as enterprise data warehousing.

7.6.3 Key Characteristics of an Enterprise Architect

7.6.3.1 Skills and Experience in Producing Designs

An enterprise architect must be proficient in the techniques that go into producing designs of complex systems, including requirements discovery and analysis, formulation of solution context, identification of solution alternatives and their assessment, technology selection, and design configuration.

7.6.3.2 Extensive Technical Breadth, with Technical Depth in One or a Few Disciplines

An enterprise architect should possess an extensive technical breadth through experience in the IT industry. This breadth should be in areas of application development and deployment, and in the areas of creation and maintenance of the infrastructure to support the complex application environment. Current IT environments are heterogeneous by nature, and the experienced enterprise architect will have skills across multiple platforms, including distributed systems and traditional mainframe environments. Enterprise architects will have, as a result of their careers, skills in at least one discipline that is considered to be at the level of a subject matter expert.

7.6.3.3 Method-Driven Approach to Execution

Enterprise architects approach their job through the consistent use of recognized design methods such as the TOGAF Architecture Development Method (ADM). Enterprise architects should have working knowledge of more than one design method and be comfortable deploying parts of methods appropriate to the situation in which they are working working. This should be seen in the body of design work the enterprise architect has produced through repeated successful use of more than one design method. Proficiency in methodology use is in knowing what parts of methods to use in a given situation, and what methods not to use.

7.6.3.4 Full Project Scope Experience

While enterprise architects are responsible for design and hand-off of the project to implementors, it is vital that they have experience with all aspects of a project from design through development, testing, implementation, and production. This scope of experience will serve to keep enterprise architects grounded in the notion of fitness-for-purpose and the practical nature of system implementation. The impact of full project scope experience should lead the enterprise architect to make better design decisions, and better inform the trade-offs made in those decisions.

7.6.3.5 Leadership

Communication and team building are key to the successful role of the enterprise architect. The mix of good technical skill and the ability to lead are crucial to the job. The enterprise architect should be viewed as a leader in the enterprise by the IT organization, the clients they serve, and management.

7.6.3.6 Personal and Professional Skills

The enterprise architect must have strong communications and relationship skills. A major task of the enterprise architect is to communicate complex technical information to all stakeholders of the project, including those who do not have a technical background. Strong negotiation and problem-solving skills are also required. The enterprise architect must work with the project management team to make decisions in a timely manner to keep projects on track.

7.6.3.7 Skills and Experience in One or More Industries

Industry skill and experience will make the task of gathering requirements and deciding priorities easier and more effective for the enterprise architect. Enterprise architects must understand the business processes of the enterprise in which they work, and how those processes work with other peer enterprises in the industry. They should also be able to spot key trends and correct flawed processes, giving the IT organization the capability to lead the enterprise, not just respond to requests. The mission of the enterprise architect is strategic technical leadership.

7.7 Conclusions

The TOGAF Architecture Skills Framework provides an assessment of the skills required to deliver a successful enterprise architecture.

It is hoped that the provision of this Architecture Skills Framework will help reduce the time, cost, and risk involved in training, recruiting, and managing IT architecture professionals, and at the same time enable and encourage more organizations to institute an internal IT architecture practice, hopefully based on (or at least leveraging) the role and skill definitions provided.