

**What is TOGAF?**

* an architecture framework.
* Tool for assisting the acceptance of EA
* Based on iterative model supported by best practices and a re-usable set of architectural assets

**TOGAF Document**

1. Introduction
2. ADM:
3. ADM Guidelines
4. Architecture Content Framework: metamodel for artifacts ABBs, overview of deliverables
5. Enterprise Continuum and Tools: Tools to categorize and store outputs of architecture activity
6. TOGAF Reference Models: TRM and III-RM
7. Architecture Capability Framework: process to establish architecture practice

**Architecture:**

* formal description of a system or a detail plan of system at a component level.
* Components, their relationships and the principles governing their design

**Enterprise/ Why Enterprise Architecture**

* Organizations that have a common set of goals, including partners
* Lower business operations costs, more agile, more flexible workforce
* Lower software cost, increase portability, improve ability to address security
* Reduce complexity on IT, maximum ROI, flexibility to make or buy
* Faster procurement process, support multi-vendor

**Architecture Framework/ Why TOGAF**

* A foundation structure, or set of structure to develop a broad range of architectures
* A method for designing a target state of enterprise in terms of BB and how they fit
* Include a set of tools and common vocab, a list of recommended standards and compliant products to implement BB.
* TOGAF results in EA that is consistent, reflect stakeholder’s need, employs best practice
* Standardize and reduce risk in the architecture development process
* Provides a best practice framework for add value, enables organization to build workable solns.

**TOGAF architectures**

**Business Architecture:**  business strategy, governance, organization

**Data Architecture:** logical and physical data assets, data management

**Application Architecture:** application deployment and relationships to the core business process

**Technology Architecture:** logical h/w and s/w to support deployment of BDA

**ADM**

Preliminary phase:

Phase A: Architecture Vision:

Phase B: Business Architecture:

Phase C: Information System Architecture:

Phase D: Technology Architecture

Phase E: Opportunities and Solutions:

Phase F: Migration Planning:

Phase G: Implementation Governance:

Phase H: Architecture Change Management:

Requirements Management

ACF uses 3 categories to describe the type of architectural work product

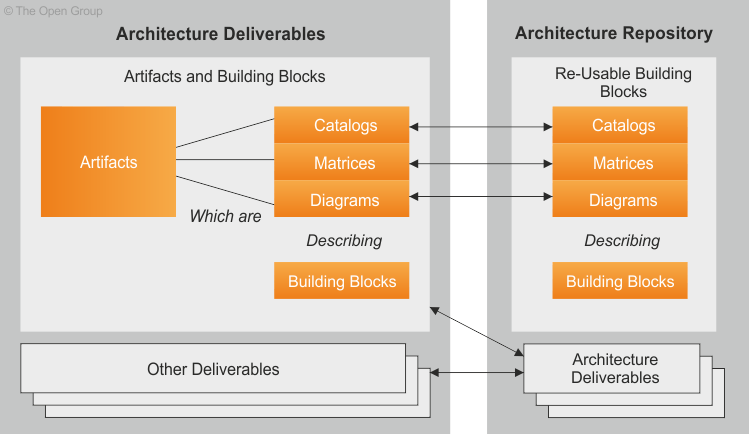
**Deliverable:** work product that is the output of the projects,

**Artifact:** architectural work product that describes an aspect of the architecture. Classify as catalogs (lists of things, requirement catalog), matrices (relationship between things, interaction matrix) and diagrams (picture of things, use-case diagram)

**Building blocks:**  a re-usable component of business / IT or architectural capability

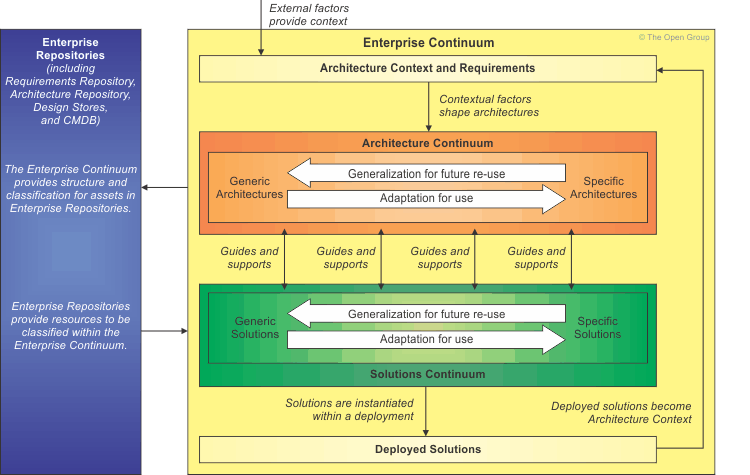
**ABBs:** describe the required capability and specification of the SBB

**SBBS:** components that will be used to implement the capability



**Enterprise Continuum**

A model for structuring a virtual repository, a method for classifying architecture and solution artifacts, show how generic solutions can be specialized from generic from orgnanization specific. Contains Architecture and Solutions Continuum.



**Architecture Repository**

Supports Enterprise Continuum by storing different classes of ADM architectural outputs

Architecture Metamodel: describes the organizational tailored application

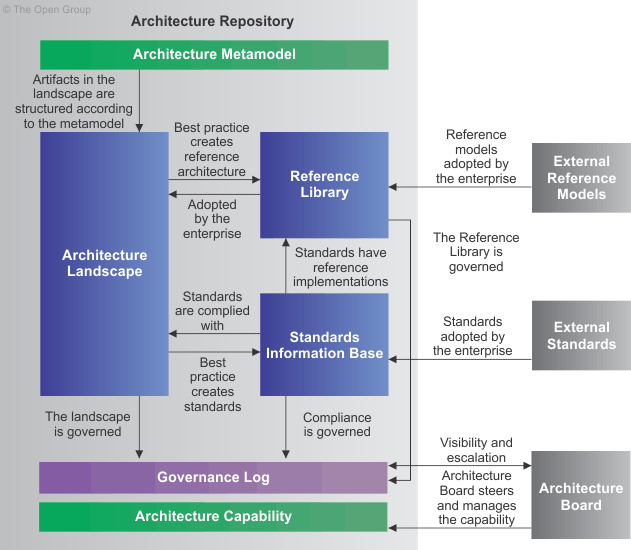
Architecture Landscape: architectural rep. of the assets at a certain time

Reference Library: guidelines to help accelerate creation of new architectures

Standard Information Base: standard for architecture to comply

Governance Log: record of governance activity

Architecture Capability: defines the structure and processes that supports the AR



TOGAF is flexible and adaptable. Two of the key elements of any enterprise architecture framework are:

* A definition of the deliverables that the architecting activity should produce
* A description of the method by which this should be done

**Maintain an Enterprise Architecture Capability**

* Establish an architecture capability**,** Architecture Board, Architecture Compliance**,** Architecture Contracts**,** Architecture Governance, Architecture Maturity Models**,** Architecture Skills Framework

**Establish an Operational Architecture Capability**

An enterprise architecture practice must be run like any other operational unit within a business. An enterprise architecture practice should establish capabilities in the following areas:

•      Financial Management •      Performance Management Risk Management

Central to this notion is Architecture Governance, which increase transparency, control risk management, value creation through monitoring.

Within the model, the content of the TOGAF document is categorized according to the following four categories:

**TOGAF Core** consists of the fundamental concepts that form the essence of TOGAF.

**TOGAF Mandated** consists of the normative parts of the TOGAF specification. These elements of TOGAF are central to its usage and without them the framework would not be recognizably TOGAF. Strong consideration must be given to these elements when applying TOGAF.

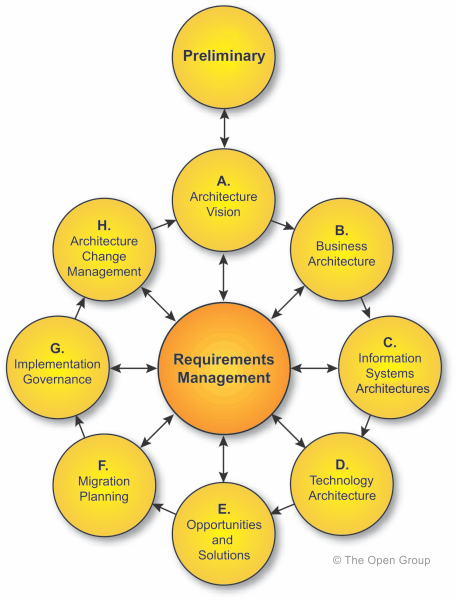
**TOGAF Recommended** consists of a pool of resources that are specifically referenced in TOGAF as ways in which the TOGAF Core and Mandated processes can be accomplished (e.g., the SEI Architecture Trade-Off Analysis Method or business scenarios).

**TOGAF Supporting** consists of additional resources that are not referenced in the other three TOGAF categories itself but provide valuable assistance.

**ADM:**

**ADM, Enterprise Continuum and Architecture Repository:**

* Practical implementation of the Enterprise Continuum will take the form of an Architecture Repository
* At relevant places through ADM, there are reminders to consider which architecture assets from Architecture Repository should use.
  + i.e. when developing Business Architecture, use a reference model from e-commerce
* Each iteration of ADM will populate an organization-specific landscape with architecture assets defined and used throughout the process.
* ADM populates the Architecture Repository with re-usable building blocks from the generic side of Enterprise Continuum.
* 1st execution of ADM is always the hardest because it has fewest re-usable assets.



The phases of the ADM cycle are further divided into steps; for example, the steps within the architecture development phases (B, C, D) are as follows:

* Select reference models, viewpoints, and tools
* Develop Baseline Architecture Description
* Develop Target Architecture Description
* Perform gap analysis
* Define candidate roadmap components
* Resolve impacts across the Architecture Landscape
* Conduct formal stakeholder review
* Finalize the Architecture
* Create Architecture Definition Document

**ADM and the Foundation Architecture:**

ADM is useful for populating Foundation Architecture of an enterprise. Business requirements can identify definitions and selections in the Foundation Architecture.

**ADM and Supporting Guidelines and Techniques:**

Describe in a separate part of TOGAF that can be referenced from the relevant points in ADM

Guidelines: describe how ADM can be adapted to deal with # of usage scenarios and special architectures (i.e. security)

Techniques: in TOGAF9 Part 3, to support application of ADM (i.e. gap analysis)

**Scoping ADM**

ADM is Iterative, over the whole process, between phases and within phases. For each iteration, Need to consider the

* Breadth of coverage:
  + what creates value to the enterprise, and select the horizontal /vertical scope, accordingly, and adjust the ADM phases accordingly
* Depth: Level of detail
* Extent of time period
* Architecture Domains: In reality not all 4 domains (BDAT) are always defined
* Architectural assets: how much can be re-used from previous iterations or other frameworks

Scope is 1st expressed in breadth, depth, and time. Then a suitable combination of domains is selected.

**Version:** illustrates the evolution of the baseline and target architecture definitions.

* Version 0.1 indicates high-level outline, Version 1.0 indicates a formally reviewed, detailed architecture

**How to adapt ADM?**

* Order of phases depends on the maturity of the enterprise. For example, if the business case of doing architecture is not well-known, then creating an Architecture vision is important
* Order of phases defined by the architecture / business principles of an enterprise. For ex., if the business process needs adjustment after its solution is packages, then Business Architecture can follow completion of Information Systems Architecture or Technology Architecture.
* Enterprise might customize ADM with another EA framework.
* ADM is mandated for use in outsourcing, need to customize to compromise between enterprise and contractor’s requirements.
* Enterprise size is large or small

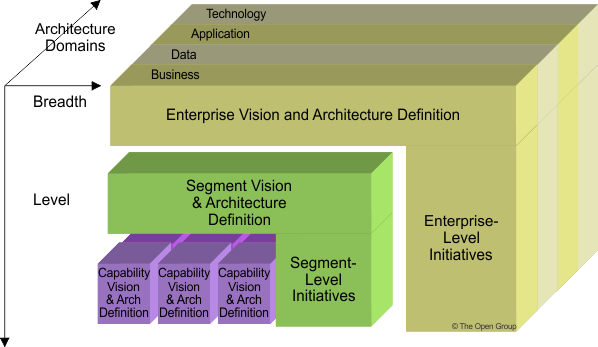
**Architecture Governance**

Compliance with ADM is important to Architecture Governance, to ensure all considerations are made and all deliverables are produced. Major information areas managed by governance repository:

* Reference Data: collateral from the Enterprise Continuum, used for guidance and instruction during project implementation
* Process Status: i.e. outstanding compliance requests, compliance assessments investigation
* Audit Information: record completed governance process actions

**Integration Framework**

A consistent frame of reference (level of details, architecture domain) must be addressed when considering integration of different architectures. At present, architecture integration that can only be accomplished at the lower end of the integration



**Preliminary phases:**

To determine and establish the architecture capability (AC)

Determine:

* Review the organization context for conducting EA
* Identify and scope the elements of the organizations affected by AC
* Identify the frameworks, methods, and process that intersect with AC
* Establish a CMM target

Establish:

* Organizational model for EA
* Detailed process and resources for architecture governance
* Select the implement tools to support AC
* Define the architecture principles

Preliminary Phase is about defining "where, what, why, who, and how we do architecture" in the enterprise. It should determine the desired approach to partition the architecture, and may be revisited in the Architecture Vision phase to ensure the AC is suitable

* Define the enterprise
* Identify key drivers and elements in organizational context
* Define requirements for architecture work
* Define architecture principles that will inform any architecture work and form the constraint
* Define the framework to be used
* Define relationships between management frameworks
* Evaluate enterprise’s maturity

**Enterprise Scope**

The scope, and whether it is federated, will determine the stakeholders who derive most benefit from AC. It is imperative to appoint a sponsor.

**Organizational context**

What’s the context surrounding the architecture? This includes

* Stakeholders, Intentions and culture of the organization, Skills and capabilities of the enterprise
* Models and budget for the EA, current process, BA

Review of organization provide requirements on how to tailor the architecture framework in terms of

* Level of formality, level of expenditure, touch-points with other organizations

**Requirements for Architecture Work**

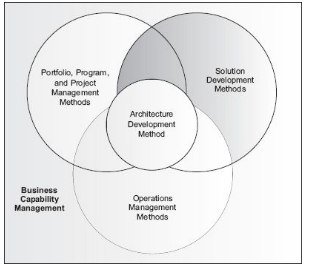
1 of these requirements needs to be articulated to define and establish Architecture Capability

* Business requirements, cultural aspirations, organization intents, strategic intents, forecast financial requirements

**Management Framework**

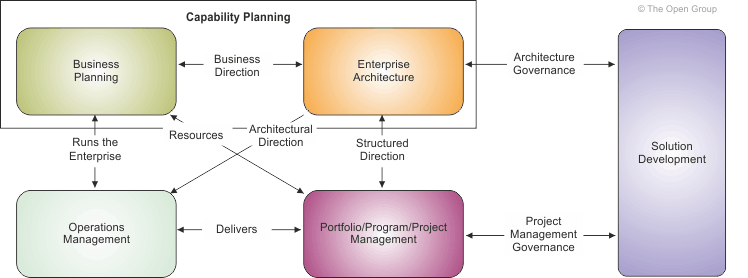
The main frameworks suggested to co-ordinate with TOGAF. These frameworks overlap with BCM, must be aware of the architecture’s impact on entire enterprise.

* **Business Capability Management** what business capabilities are required to deliver business value
* **Portfolio/Project Management Methods** how a company manages its change initiatives.
* **Operations Management Methods** how a company runs its day-to-day operations
* **Solution Development Methods** how business systems are delivered in accordance IT



The frameworks complement each other.

* Business planning provides initial direction to EA.
* Solution development (*developers the components*) is used within the Portfolio Management Framework to create the architectural component specified.
* Project management (*delivers the components*) receives the direction to build required.
* Operation management (*incorporates the components*) receives the deliverable and integrates them with the corporate infrastructure.



**Inputs to Preliminary Phase**

Reference:

* TOGAF

Non-Architectural Inputs:

* Business strategy, architecture capability, contract agreements, major frameworks like Portfolio management

Architectural Inputs:

* organizational model such as CMM, existing architecture framework

**Steps to Preliminary Phase**

1. Scope the organizations impacted:
   1. Identify core enterprise/soft enterprise/extended enterprise /communities involved/governance
2. Confirm Governance and Support Frameworks:
   1. Understand how guidelines, models, is brought up by governance
   2. Likely existing governance and support models will need to change to support new architecture framework. The impacts needs to be understood and agreed by stakeholders
3. Define EA team and organization:
   1. Determine existing capability, assess architecture maturity, determine constraints
4. Identify and Establish Architecture Principles
5. Tailor TOGAF
   1. Consider the need for terminology, process, content
6. Implement Architecture Tools

**Outputs of Preliminary Phase**

1. Organizational Model for EA
2. Tailored Architecture Framework
3. Initial Architecture Repository
4. Request for Architecture Work
5. Architecture Governance Framework
6. Might include catalogs: principles catalog

**Phase A: Architecture Vision**

Objectives:

1. Develop a high-level aspirational vision of capabilities and business value to be delivered by the EA
   1. Ensure proper endorsement from corporate, and support from line management
2. Obtain approval for Statement of Architecture Work

Creating the Architecture Vision

* + Describes how the new capability will meet the business goals and strategic objectives
  + To verify and understand the documented business strategy and goals, and may integrate enterprise strategy and goals with current architecture
  + Provides a 1st cut, high-level description of the Baseline and Target Architectures, covering BDAT
  + Once an architecture vision is defined, use it to build a consensus

Inputs:

* Non-Architectural Inputs:
  + Request for Architecture Work
  + Business principles, business goals
* Architectural Inputs:
  + Organizational model
    - Scope of organization impact; Maturity; Constraints
  + Tailor Architecture Framework
  + Populated Architecture Repository
* Steps
  + Establish the Architecture Project
  + Identify Stakeholders, Concerns and Business Requirements
    - Identify candidate vision and requirements
    - Identify candidate scope boundaries
    - Identify stakeholder concerns issues
  + Confirm and Elaborate Business Goals, Business Drivers and Constraints
    - Identify the business goals and strategic drivers of the organization
  + Evaluate Business Capabilities
    - Capability for enterprise to develop and consume architecture
      * Gaps identified in enterprise capability to execute on change will inform the architect on the description of the Target Architecture and on the Implementation and Migration Plan
    - Baseline and target capability level of the enterprise
      * Gaps identified in Architecture Capability require iterations between Architecture Vision and Preliminary Phase to ensure Architecture Capability addresses scope
  + Assess Readiness for Business Transformation
    - Results are added to the capability assessment
  + Define Scope
    - What is in and out of the Baseline Architecture and Target Architecture
    - Breadth, level of detail, partitioning characteristics, specific architecture domains to be covered, time period, architecture assets to be leveraged
  + Confirm and Elaborate Architecture Principles, including Business Principles
    - Review the principles for developing the architecture. Architecture principles are normally based on principles developed from Preliminary Phase.
  + Developer Architecture Vision
    - Covers breadth of scope of project, drawing a simple solution concept
    - The 1st, high-level definition of the baseline and target environment
    - Initial version should be stored in the Architecture Repository
  + Define the Target Architecture Value Propositions and KPIs
    - Develop the business case and value proposition
  + Identify the Business Transformation Risks and Mitigation Activities
    - Initial level of risk and residual level of risk
  + Developer SoA Work, secure approval

Outputs

* Approved SoA
  + Architecture vision overview, project plan, scope
* Architecture Vision
  + Problem description, Objective, Summary views, Refined key stakeholders agreements
* Refined business principles statements
* Architecture principles
* Capability Assessment
* Tailored Architecture Framework
* Draft ADD 0.1
* Communication Plan, Additional Content for Architecture Repository

**Phase B: Business Architecture**

Objective:

1. Develop the target business architecture that describes how the enterprise needs to operate to achieve the business goals, and respond to the strategic drivers set out in the Architecture Vision, and addresses the Request for Architecture Work and stakeholder concerns
2. Identify candidate Architecture Roadmap components based on gaps between baseline and Target Business Architecture

Overview:

* Business Architecture: describes the product strategy and the organizational, functional, process information, and geographic aspects of the business environment.
* Knowledge of BA is a prerequisite for architecture in Data, Application, and Technology. Therefore it is the 1st architecture activity to be undertaken.

Develop baseline description:

* In Baseline Description, however, the analysis is done from bottom-up. Architect has to document the working assumptions about high-level architectures

Business Modeling:

To map from high-level business requirements down to more detailed ones. Sample techniques

* Activity Models: describes the function of business activities, and data/information exchanged
* Use-Case Models: describes the business process
* Class Models: describes relationship between information
* Node Connectivity Diagram/Information Exchange Matrix: used by defense

Using the Architecture Repository:

The Architecture team will need to consider what relevant BA resources are available from the Architecture Repository

* Industry Architectures: generic, relevant to the industry sector
* Common Systems Architectures: relevant to common high-level business domains
* Enterprise-specific building blocks: process components, business rules
* Applicable standards

Inputs

* Organizational Model for the EA
* Tailored Architecture Framework
* Approved SoA Work
* Architecture principles/Enterprise Continuum
* Architecture Repository
* Architecture Vision
* Draft ADD including Baseline BA/TA/DA/AA, Target BA/TA/DA/AA

Steps

* Select Reference Models Viewpoints *(e.g. operations, management, financial)* and Tools
  + Determine overall modeling process
  + Techniques such as Activity models, use-case models, and class models
    - * Structured Analysis (identify key business functions) / Use-case Analysis (break-down across actors and organization)/ Process Modeling (identify elements of the process)
  + Identify require service granularity level, boundaries, & contracts
  + Identify required catalogs of Business building blocks
  + Identify required matrices
  + Identify required diagrams
  + Identify the types of requirements to be collected
* Develop Baseline Business Architecture Description
* Develop Target Business Architecture Description
* Perform Gap Analysis, Define Candidate Roadmap Components
* Resolve impact across landscape, Conduct formal stakeholder review
* Finalize Business Architecture, Create ADD

Output

* Refined Architecture Vision phase deliverables => updated SoW, architecture principles
* Draft ADD, including
  + Baseline Business Architecture (1.0), Views corresponding to stakeholder concerns
  + Target Business Architecture (1.0): business functions breakdown, organization structure, business goals and objectives for each unit, business service provide to customers, processes, roles, data model
* Draft Architecture Requirements Specification, including BA requirements such as:
  + gap analysis results, updated business requirements, technical requirements
* Business Architecture components of an Architecture roadmap

**Phase C: Information Systems Architecture**

Objectives:

Develop the Target Data Architecture that enables the Business Architecture and Architecture Vision, while addressing the Request for Architecture Work and stakeholder concerns.

Identify candidate Architecture Roadmap components based upon gaps between Baseline and Target Data Architecture.

Key considerations:

* Data Management
  + Define which application components serve as system of record for enterprise master data
  + Will there be enterprise-wide standards that all application components need to adopt?
  + Understand how data entities are used by business functions, process, services
  + Understand how and where enterprise data entities are created, stored
  + What is the level and complexity of data transformations for information exchange?
  + Define the requirement for software to support data integration with customers/suppliers
* Data Migration
  + Identify data migration requirements to
    - ensure the target application has quality data when it is populated
    - Enterprise-wide common data definition is established to support the transformation
* Data Governance: Ensure the enterprise has necessary dimensions in place to enable transformation
  + Structure: to manage the data entity aspects of the transformation
  + Management system: to manage the governance aspects of data entities
  + People: resource and skills require for the transformation

Architecture Repository:

ARTS has a data model for Retail industry, Energistics have a data model for Petro-technical industry

Inputs:

Non-Architectural Inputs: Request for Architecture Work, Capability Assessment, Communication Plan

Architectural Inputs:

* Organization Model for EA
* Tailored Architecture Framework
* Data principles, Application principles
* SoA Work, Architecture Vision,
* Architecture Repository (reference models, re-usable building blocks for definition of data)
* Draft ADD
  + Baseline / Target Business Architecture (1.0)
  + Baseline / Target Data Architecture (0.1)
  + Baseline / Target Application Architecture (1.0 or 0.1)
  + Baseline/ Target Technology Architecture (0.1)
* Draft ARS
  + Gap analysis results, relevant technical requirements that will apply
* Business Architecture components of an Architecture Roadmap

Steps:

* Select reference models, viewpoints, and tools
  + Review and validate the data principles
  + Select relevant Data architecture resources (i.e. reference models, patterns)
  + Select relevant Data architecture viewpoints (i.e. address concerns from stakeholders), identify the tools and techniques (Entity-relationship diagram / Class diagrams)
  + Determine overall modeling process
    - Collect data-related models from existing Business & Application Architecture materials
    - Rationalize data requirements and align with existing enterprise data catalogs
    - Update and develop matrices across the architecture
    - Elaborate data architecture views by examining how data is created, distributed, migrated, secured and archived.
  + Identify required catalogs of Data Building Blocks
    - Data inventory is captured as a catalog within the Architecture Repository
    - A business service / information diagram was created showing key data entities required by main business services
    - Data Entity / Data Component catalog should be consider for development
  + Identify required matrices
    - To show the core relationships between related model entities. Matrices like
      * Data entity / Business Function (show which data support which function, which business function owns which data)
      * Business Services / Information (developed during the Business Architecture phase)
      * Application / Data (developed across the Application and Data phase)
  + Identify required diagrams
    - A diagram of relationships between entities and their attributes
      * Level of detail varies for each data model
    - Conceptual data diagram, logical data diagram, data dissemination diagram, data lifecycle diagram, data security diagram, data migration diagram
  + Identify types of requirement to be collected
    - Collect requirement related to the data domain
    - Requirement input into the Application + Technology Architectures
    - Detailed guidance to be reflected during the design and implementation
* Develop Baseline Data Architecture Description
* Develop Target Data Architecture Description
  + To the extent to support the Architecture Vision and Target BA. Scope and detail depends on the relevance of the data elements , and if architectural descriptions exist
* Perform Gap Analysis
  + Trade-off analysis to resolve conflicts among different views
  + Validate the models support the principles, objectives and constraints
  + Test architecture models for completeness vs. requirements
  + Document changes to the viewpoints represented in selected models from Repository
* Define candidate roadmap components
* Resolve impacts across the architecture landscape
  + Understand any wider impact or implications
* Conduct formal stakeholder review
  + Identify areas where the Application Architecture may need to change to cater for changes in the Data Architecture
* Finalize the data architecture
  + Select standards for each building blocks, and re-use from reference models
* Create Architecture Definition Document
  + Document rationale for building block decisions in the ADD
  + Prepare Data Architecture sections of the ADD with the following models
    - Business data, logical data, data management process, data entity / business function, data interoperability requirements

Outputs:

* Updated Architecture Vision phase deliverables, such as SoA work update, updated data principles
* Draft ADD including
  + Base/Target data architecture 1.0 (business/logical data model, data management process models, data entity/business function matrix)
  + views corresponding to the selected viewpoints addressing stakeholders concerns
* Draft ARS with gap analysis results, data interoperability requirements, etc.
* Data Architecture components of an Architecture Roadmap
* Data Entity / Data Component catalogs
* Data Entity / business function matrix, Application / Data matrix
* Diagrams for Conceptual Data, Logical Data, Data Dissemination, Data security, Data Migration

**Phase C: Information Systems Architecture – Application Architecture**

* Develop the target application architecture that enables Business Architecture and Architecture Vision, while addressing the Request for Architecture Work and stakeholder concerns

Approach

What relevant application architecture is available in the Architecture Repository?

Generic business models: TMF (TeleManagement Firm), OMG (Object Management Group) for specific healthcare, finance etc.

Open Group’s Integrated Information Infrastructure (III-RM) focus on application-level components and service to provide an integrated information infrastructure.

Inputs

* Reference materials external to Enterprise
* Request for Architecture Work, Capability Assessment, Communication Plans
* Tailored Architecture Framework
* Application principles
* Architecture Repository, including re-usable building blocks, reference models, organization standard
* Draft ADD, baseline/target BA (1.0), baseline/target DA (1.0 or 0.1), baseline/target AA if available (0.1), baseline/target TA (0.1) vision
* Draft ARS, including Gap analysis
* Business and data architecture components of an Architecture Roadmap

Steps

Select Reference Models, Viewpoints and Tools

* Determine Overall Modeling Process
  + Understand the list of applications that are required
  + Decompose the complicated application into more than 1 application
  + Ensure the set of application definitions is internally consistent
  + Develop matrices across architecture by relating applications to business service, business function, data, process
  + Elaborate a set of Application Architecture views
* Identify Required Catalogs of Application Building Blocks
  + Application Portfolio is captured as a catalog within the Architecture Repository. Catalog is hierarchical and captures the decomposition across model entities (e.g. logical -> physical -? Information system service)
  + Use Application Portfolio catalog , Interface catalog
* Identify Required Matrices
  + Once applications are mapped to business services, it is possible to make association from applications to data, through the business information diagrams developed during Business Architecture
  + Application architecture should identify user and organizational dependencies on applications
  + Application/Organization matrix, Role/Application matrix
* Identify Required Diagrams
  + Application Communication diagram, Application and user location diagram, Application Use-case diagram
* Identify types of requirements to be collected
  + Formalize the application-focused requirements for implement Target Architecture

Develop Baseline Application Architecture Description

Develop Target Application Architecture Description

Perform Gap Analysis

Define candidate roadmap

Resolve impact across the architecture landscape

Conduct formal stakeholder review

Finalize the Application Architecture

* Select standards for each building block, and document them, Document cross-check of overall architecture against business requirements, document rationale for building block decisions

Create ADD

Outputs

Refined Architecture Vision phase deliverables: updated SoA work, validated or new App principles

Draft ADD:

* Baseline Application Architecture 1.0
* Target Application Architecture 1.0
  + Process system model, Place systems model, Time systems model, People systems models
* Draft Architecture Requirement Specification including Application Architecture requirements
* Application Architecture components of an Architecture Roadmap

**Phase D: Technology Architecture**

Develop the Target Technology Architecture that enables the logical and physical application and data components and Architecture Vision, addressing Request of Architecture Work & stakeholder concerns.

Identify candidate Architecture Roadmap components based upon gaps between the Baseline and Target Technology Architectures

Approach

Architecture Repository: Re-use existing IT services, TOGAF TRM, Generic technology models (TMF),

Inputs

Request for Architecture Work, Capability Assessment, Communication Plans

Organizational Model, Tailor Architecture Framework, Technology principles, SoA Work,

Architecture Vision, Architecture Repository (Building blocks, reference models)

Draft ADD, Draft ARS,

Business/Data/Application Architecture components of an Architecture Roadmap

Steps

Select Reference Models, Viewpoints and Tools: review technology principles, select TA resources

Determine Overall Modeling Process:

* Define a taxonomy of platform services
* Identify relevant locations where technology is deployed
* Carry out a physical inventory of deployed technology and abstract up to fit into the taxonomy
* Is the technology in place to meet new requirement?
* Determine impact (sizing and cost, capacity planning, installation/governance)

In earlier ADM phases, decision on service granularity will impact technology component, such as

* Performance (more coarse-grain, slower),
* Maintainability (more coarse-grain, harder to maintain),
* Location and Latency (for inter-service communication, need to consider service boundary for location impact) and
* Communication Availability (need to consider during service decomposition)

Identify Required Catalogs of Technology Building Blocks:

Catalogs form the raw material for development of matrices and diagrams, and also act as key resource for portfolio managing business and IT capability. Ex. Technology Standards / Technology Portfolio catalog

* Collect a list of products in use
* If the requirements identified in the Application Architecture are not met by existing products, extend the product list
* Classify products against the TOGAF TRM. If technology standards are currently in place, apply these to the technology component catalog to gain a baseline view of compliance

Identify Required Matrices:

Matrices show the core relationships between related model entities. Ex. Application/Technology matrix

Identify Required Diagrams

* Diagrams present the TA information from a set of different perspectives.
* For major base-line application or application platforms, product a stack diagram showing how hardware, OS, software and packaged applications combine.
* For each environment, produce a logical diagram of hardware and software infrastructure to show the logical communications.
* For each environment, produce physical diagram of communication infrastructure, such as routers, switches, firewalls.

Identify Types of Requirement to be collected

* Once the TA catalogs, matrices, and diagrams have been developed, formalize the technology-focused requirements to complete architecture modeling.

Select Services

The service portfolios are combination of basic services from the service categories in the TOGAF TRM that do not conflict. The requirements identified can provide information about

Requirements for organization-specific elements or pre-existing decisions

Pre-existing and unchanging organizational elements

Inherited external environment constraints

For each building block, build up a service description portfolio, and the set must be tested to ensure the functionality provided meets application requirements.

Develop Baseline Technology Architecture Description

Develop Target Technology Architecture Description

Perform Gap Analysis

Define Candidate Roadmap Components

Resolve Impacts across Architecture Landscape

Conduct Stakeholder Review

Finalize Technology Architecture

Create ADD

Outputs

Draft ADD

* Baseline TA
* Target TA 1.0 with
  + Technology components and their relationship to information systems
  + Technology platforms and their decomposition
  + Environments and locations
  + Hardware and Network specification
* Views addressing key stakeholder concerns

Draft ARS (Gap analysis results, requirements output from Phase B and C, technology requirements)

Technology Architecture components of an Architecture Roadmap

**Phase E: Opportunities and Solutions**

Generate the initial completion version of the Architecture Roadmap, based upon the gap analysis and candidate Architecture Roadmap components from Phases B,C,D.

Determine whether an incremental approach is required, and if so, identify the Transition Architectures

**Approach**

Concentrate on how to deliver the architecture. Provides basis of creation of Implementation and Migration Plan which is completed in Phase F.

4 concepts to transition from developing to delivering a Target Architecture:

* Architecture Roadmap: lists individual work packages to realize TA
* Work Packages: logical group of changes to realize TA
* Transition Architectures: An interim TA that describes the enterprise at an architecturally significant state between BA and TA.
* Implementation and Migration Planning: provides a schedule of projects

**Inputs**

Organizational Model for EA

Governance model and frameworks for EA, Business Planning, etc.

Tailored Architecture Framework including Architecture method and content

SoA Work, Architecture Vision

Architecture Repository, including re-usable BB, reference models

Draft ADD (all version 1.0, detailed)

Draft ARS, including Architectural requirements, Gap analysis, IT service management requirements

**Steps**

* Determine/Confirm key corporate change attributes
  + Creation of Implementation Factor Assessment and Deduction matrix to serve as repository for architecture implementation and migration decisions, document the transition capabilities of organization unit involve and enterprise culture/skills
* Determine Business Constraints for Implementation
  + Review business plan / Enterprise Maturity to find business drivers that constrain implementation
* Review and Consolidate Gap Analysis Results from Phases B to D
  + Create a Consolidated Gaps, Solutions and Dependencies matrix to identify Solution BB
  + Use Business Interaction matrix, Data Entity/Business Function matrix and Application/Function matrix to relate elements from different architectural domains
  + Re-organize the gaps, refer to the Implementation Factors Assessment and Deduction matrix
* Review and Consolidate Requirements Across Related Business Functions
  + Assessment the requirements, gaps, solutions and factors to identify the minimal requirements
* Consolidate and Reconcile Interoperability Requirements
  + Consolidate the requirements from previous phase. Review and consolidate the Architecture Vision, TA, Implementation Factor Assessment Matrix, Consolidated Gaps matrix
  + Key outcome is minimize interoperability conflicts; either create building block that transforms or translates conflicting building block, or change specification of a conflicting building block
* Refine and validate dependencies
* Confirm Readiness and Risk for Business Transformation
* Formulate Implementation and Migration Strategy
  + Determine the overall strategic approach to implement the solutions
    - Greenfield: completely new implementation
    - Revolutionary: radical change
    - Evolutionary: convergence, parallel running or phased approach
  + Determine the approach to address and mitigate risk identified in the Consolidated Gaps, Solutions and Dependencies matrix
    - Quick win
    - Achievable targets
    - Value chain method
  + These approaches and identified dependencies form basis for creation of work packages
* Identify and Group major work packages
  + Fill in the ‘Solution” column in the Consolidated Gaps, Solutions and Dependencies matrix. Indicate for every gap whether the solution is new development, or on existing product
  + Decompose top-level work packages into increments
  + Group work packages into portfolios
* Identify Transition Architectures
* Create Architecture Roadmap and Implementation and Migration Plan
* Outputs
  + Architecture Roadmap, including
    - Work package description, Functional requirements, Dependencies, Implementation Factor Assessment and Deduction Matrix
  + Implementation recommendations
    - Criteria measures of effectiveness, risks and issues, Solution BBs
  + Implementation and Migration plan and strategy (0.1)

**Phase F: Migration Planning**

Objectives:

* Ensure the Implementation and Migration plan is coordinated with the enterprise’s approach to manage change
* Finalize the Architecture Roadmap and the Supporting Implementation and Migration Plan
* Ensure the business value and the cost of work packages and Transition Architectures is understood by key stakeholders

Approach:

Incomplete Architecture Roadmap and Implementation/Migration Plan (V0.1) from Phase E are integrated in Phase F.

Assess the dependencies, costs, benefits of various migration projects

Architecture Development cycle should be completed, and lessons learnt documented

Steps:

* Confirm management framework interactions for the Implementation and Migration Plan
  + Business Planning, Enterprise Architecture, Portfolio/Project Management, Operations Management
* Assign a business value to each work package
  + Estimate the business value of each project using the Business Value Assessment Technique
  + Performance Evaluation Criteria, Return-on-Investment Criteria, Business Value, Critical Success Factors, Measure of Effectiveness, Strategic fit
* Estimate Resource Requirements, project timings and availability/delivery vehicle
* Prioritize the Migration Projects through the conduct of a Cost/Benefit assessment and risk validation
* Common Architecture Roadmap and Update ADD
* Generate the Implementation and Migration Plan
* Complete the Architecture Development Cycle and Document Lessons Learnt

Outputs:

Implementation and Migration Plan (V1.0)

* Implementation and Migration Strategy
* Project and portfolio breakdown of the implementation:
  + Allocation of work packages to project and portfolio / capabilities delivered by projects / WBS / Milestones and timing

**Phase G: Implementation Governance**

Objectives:

Ensure conformance with the Target Architecture by implementation projects

Perform appropriate Architecture Governance functions for the solution and any implementation-driven Architecture Change Request

Approach:

Deploy Target Architecture as transitions, and

* Establish an implementation program to enable delivery of Transition Architecture
* Adopt a phased deployment schedule
* Follow organization’s standard and established portfolio/program management approach
* Define an operations framework to ensure effective long life of deployed solution

Steps:

* Confirm scope and priority for deployment with Development Management
  + Perform gap analysis on EA and identify specify solution Building Blocks
* Identify deployment resources and skills
  + Identify system development methods and ensure the method allows feedback to architecture team on designs
* Guide Development of Solutions Deployment
  + Produce Implementation Plan
* Perform Enterprise Architecture Compliance reviews
* Implement Business and IT Operations
* Perform Post-Implementation review and close the implementation

Outputs:

* Architecture-compliant solutions deployed including
  + Implemented system
  + Populated Architecture Repository
  + Architecture compliance recommendation
  + Architecture Vision and ADD, updated post-implementation

**Phase H: Architecture Change Management**

The goal of an architecture change management process is to ensure that the architecture

achieves its original target business value. This includes managing changes to the architecture

in a cohesive and architected way.

Objectives:

Ensure Architecture lifecycle is maintained

Ensure Architecture Governance Framework is executed

Ensure the EA capability meets current requirement

Approach:

Guideline for Maintenance vs. Architecture redesign

If the change impacts 2 stakeholders (i.e. impact business strategy), likely require re-entry to ADM

If the change impacts 1 stakeholder (i.e. new standards emerge), candidate for change management

If the change is at an infrastructure level, (i.e. change to Baseline description of the Technology Architecture), candidate for change management.

Steps:

Establish value realization process

Deploy monitor tools

Manage risks

Provide analysis for Architecture Change Management

Develop change requirements to meet performance targets

Manage Governance Process

Activate the process to implement change

Outputs:

Maintenance changes for

Architecture updates, Change to architecture framework, new Request for Architecture

Simplification change to an architecture is often driven by a requirement to reduce investment; A simplification change can normally be handled via change management techniques.

Incremental change is driven by a requirement to derive additional value from existing investment; An incremental change may be capable of being handled via change management techniques, or it may require partial re-architecting

re-architecting change is driven by a requirement to increase investment in order to create new value for

exploitation. A re-architecting change requires putting the whole architecture through the architecture development cycle again.

**Architecture Requirements Management**

Objective:

Ensure requirement management process is sustained and operates for all relevant ADM phases

Management architecture requirements identified during execution of ADM cycle or phase

Ensure the relevant architecture requirements are available for use by each phase

Approach:

ADM is continuously driven by requirements management process.

Requirements management process itself does not dispose/prioritize requirements; this is done within relevant phase of ADM.

In each ADM relevant phase, architect should identify types of requirement

Steps:

* RMS: Monitor baseline requirements
* ADM: Identify changed requirements
* RMS: Identify changed requirements and record priorities
* ADM: Assess impact of changed requirements
* ADM: Implement requirements arising from Phase H
* RMS: Update requirement repository
* ADM: Implement changes in current phase, assess and revise gap analysis for past phases

Outputs

Requirements Impact Assessment

Architecture Requirements Specification

**ADM Techniques**

The following techniques are described within [Part III: ADM Guidelines and Techniques](http://pubs.opengroup.org/architecture/togaf9-doc/arch/pt3.html) to support specific tasks within the ADM:

* Architecture Principles (see [*23. Architecture Principles*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap23.html#tag_23)) - principles for the use and deployment of IT resources across the enterprise - describes how to develop the set of general rules and guidelines for the architecture being developed.
* Stakeholder Management (see [*24. Stakeholder Management*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap24.html#tag_24)) describes Stakeholder Management, an important discipline that successful architecture practitioners can use to win support for their projects.
* Architecture Patterns (see [*25. Architecture Patterns*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap25.html#tag_25)) provides guidance on using architectural patterns.
* Business Scenarios (see [*26. Business Scenarios and Business Goals*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap26.html#tag_26)) describes the Business Scenarios technique, a method for deriving business requirements for architecture and the implied technical requirements.
* Gap Analysis (see [*27. Gap Analysis*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap27.html#tag_27)) describes the technique known as gap analysis. It is widely used in the TOGAF ADM to validate an architecture that is being developed.
* Migration Planning Techniques (see [*28. Migration Planning Techniques*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap28.html#tag_28)) describes a number of techniques to support migration planning in Phases E and F.
* Interoperability Requirements (see [*29. Interoperability Requirements*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap29.html#tag_29)) describes a technique for determining interoperability requirements.
* Business Transformation Readiness Assessment (see [*30. Business Transformation Readiness Assessment*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap30.html#tag_30)) describes a technique for identifying business transformation issues.
* Risk Management (see [*31. Risk Management*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap31.html#tag_31)) describes a technique for managing risk during an architecture/business transformation project.
* Capability-Based Planning (see [*32. Capability-Based Planning*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap32.html#tag_32)) describes the technique of capability-based planning.

TOGAF is designed to be flexible and it can be used with various architectural styles. TOGAF ensures that the needs of each stakeholder are appropriately addressed in the context of other stakeholders and the Baseline Architecture. Addressing a distinctive style should not call for significant changes to TOGAF; instead it should adjust the models, viewpoints, and tools used by the practitioner.

In Phase B, Phase C, and Phase D the practitioner is expected to select the relevant architecture resources, including models, viewpoints, and tools, to properly describe the architecture domain and demonstrate that stakeholder concerns are addressed. Addressing the distinctive features will usually include extensions to the Architecture Content Metamodel and the use of specific notation or modeling techniques and the identification of viewpoints.

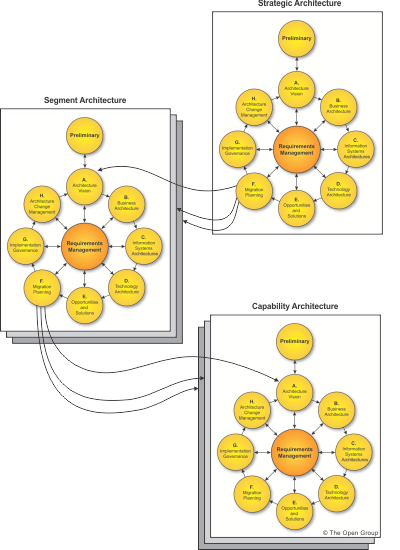
**Applying Iterations to ADM**

Two approaches can be adopted within the ADM for the development of architectures:

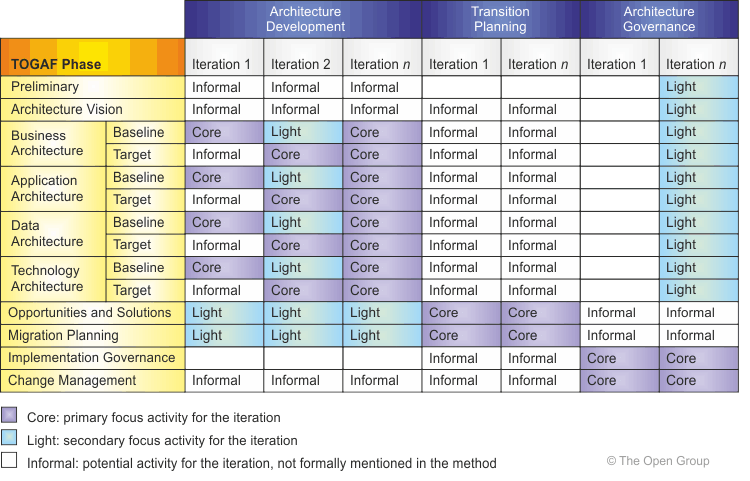
* **Baseline First**: In this style, an assessment of the baseline landscape is used to identify problem areas and improvement opportunities. This process is most suitable when the baseline is complex, not clearly understood, or agreed upon. This approach is common where organizational units have had a high degree of autonomy.
* **Target First**: In this style, the target solution is elaborated in detail and then mapped back to the baseline, in order to identify change activity. This process is suitable when a target state is agreed at a high level and where the enterprise wishes to effectively transition to the target model.

Typically, if the baseline is broadly understood a higher value will be obtained focusing on the target first then baseline to the extent necessary to identify changes.

Each iterations complete an ADM cycle at a single level of architecture description as illustrated below.

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/20_adm_hierarchy.png)

The following tables show at a high level which phases should be completed for which iteration cycle, showing activity that is core (i.e., the primary focus of the iteration), activity that is light (i.e., the secondary focus of the iteration), and activity that may be informally conducted (i.e., some activity may be carried out, but it is not explicitly mentioned in the ADM).

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/19_baseline.png)

How this iteration process is exercised is by:

* The formality and nature of established process checkpoints within the organization.
* The level of stakeholder involvement expected within the process
* The number of teams involved and the relationships between different teams.
* The maturity of the solution area and the expected amount of rework and refinement required to arrive at an acceptable solution.
* Attitude to risk.
* The class of engagement.

**Applying ADM across Architecture Landscape**

Strategic Architecture:

Organizing framework for operational and change activity, allows for direction setting at executive level

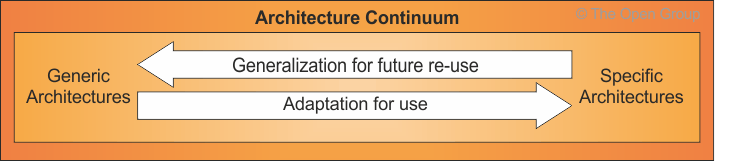
Segment Architecture:

Organizing framework for operational and change activity, allows for direction setting and development of effective architecture roadmaps at program or portfolio level

Capability Architecture:

Organizing framework for change activity and the development of effective architecture roadmaps realizing capability increments

The Architecture Continuum is a useful tool to discover commonality and eliminate unnecessary redundancy. Levels and Architecture Continuum provide a comprehensive mechanism to describe and classify the Architecture Landscape.

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/20_arch_con.png)

Organizing the Architecture Landscape

Breath: subject matter area is generally the primary organizing characteristic

Depth: broad subject areas, less details. More specific subject matter, more details

Time: broader and less detailed architecture will generally be valid for longer period of time.

Recency: each architecture view will progress through development cycle where it increases in accuracy

A number of techniques can be employed to use ADM as a process that supports such hierarchies of architectures. Essentially there are two strategies that can be applied:

1. Architectures at different levels can be developed through iterations within a single cycle of the ADM process.
2. Architectures at different levels can be developed through a hierarchy of ADM processes, executed concurrently.

**Security Architecture and ADM**

Security architecture generally has the following characteristics

Own discrete security methodology

Own discrete views and viewpoints

Calls for unique set of skills and competencies of enterprise and IT architects

Introduces unique, single-purpose components in the design

Areas of concern for security architect

Authentication:

Authorization:

Audit:

Assurance:

Availability:

Asset Protection:

Administration:

Risk Management:

Preliminary Phase:

Scope the enterprise organizations impacted by security architecture

* Identify core/soft/extended/communities/security governance involved

Define and document applicable regulatory and security policy requirements

Define the required security capability as part of Architecture Capability

Implement security architecture tools

Inputs: written security policy Outputs: list of applicable regulations/security policies

Architecture Vision:

Obtain management support for security measures

Define necessary security-related management sign-off milestones of this cycle

Determine and document applicable disaster recovery or business continuity plans/requirements

Identify and document the anticipated physical/business/regulatory environments

Determine and document the criticality of the system: safety-critical/mission-critical/non-critical

Inputs: disaster recovery and business continuity plans

Outputs: security policy, list of checkpoints, disaster recovery and business continuity plans, environment statements

Business Architecture:

Determine the legitimate actors who are interacting with the products

Assess and baseline security-specific business processes (enhancement of existing objective)

Determine what can go wrong

Determine the assets at risk if something goes wrong

Inputs: environment statements, plans

Outputs: new disaster recovery / continuity requirements, security actors, baseline/target security processes, asset list with values and owners, threat analysis matrix

Information Systems Architecture:

Baseline current security-specific architecture elements

Determine and document the sensitivity of information stored/created/used

Determine approaches to address identified risks

Identify actions/events that warrant logging for later review of triggering forensic processes

Inputs: threat analysis matrix, documented forensic processes,

Outputs: risk management strategy, security use-case models, data lifecycle definitions

Technology Architecture:

Assess and base-line current security-specific technologies

Identify minimal privileges required for any entity to achieve technical or business objective

Identify mitigating security measures, where justified by risk assessment

Inputs: risk management strategy,

Outputs: list of security technologies, risk management plan, user trust requirements,

Opportunties and Solutions

Identify existing security services available / new code and assets for re-use

Engineer mitigation measures addressing identified risks

Evaluated tested and re-usable security software

Migration Planning

Assess the impact of new security measures upon other new components or existing leveraged systems

Implement assurance methods for measuring and communicating security measures

Implement disaster recovery and business continuity plans or modifications

Implementation Governance

Establish architecture artifact, design, and code reviews and define acceptance criteria

Implement methods and procedures to review evidence produced by the system that reflects stablility

Implement necessary training

Architecture Change Management

Incorporate security-relevant changes to the environment into requirements

**TOGAF and SOA**

Strategic Architecture

* identify basic SOA issue of whether you need SOA and in which Segments
* cross-segment SOA capability requirements
* organization-specific reference architecture and high level relationships within the organization

Segment Architecture

* describe the SOA structure
* define which capabability will use SOA as an architecture style
* cross-capability and detail cross-capability relationships

Capability Architecture

* describe which services will be available
* describe SOA services that enable cross-capability re-use and capability

Preliminary Phase

Adopts service-orientation as an architecture principle. Key outputs:

Principles, Organizational structure, Governance Initial Architecture Repository

Architecture Repository

Building blocks of SOA, Detailed building blocks of SOA reference architecture, High-level perspective of SOA reference architecture,

Phase A: Architecture Vision

Stakeholders understand the implications of SOA and prepared for the organizational impacts

Phase B: Business Architecture

Use BPD, Business Vocab catalog, Business Service/Information Matrix

Produce appropriate views to demonstrate how their SOA-specific concerns are addressed

Phase C: Information Systems Architecture

Use IS Service Interaction Diagram, IS Service Contract Catalog, to demonstrate to stakeholders how their SOA-specific concerns with Applications Architecture are addressed.

In each of Phase B,C,D, gap analysis should be performed between Baseline and Target Architectures. For B,D and Data Architecture of C, not affected by SOA. For Application Architecture of C, SOA affects how gap analysis is performed. ABB defined in Phase C include traditional applications and groups of services, and both BBs should be included in gap analysis

Phase D: Technology Architecture

Defines the S/W & H/W to support portfolio services

Use Logical Technology Diagram/Technology Matrix

Phase E: Opportunities & Solutions

Identify SOA solution. Use Physical SOA solution matrix/diagram

**Architecture Principles**

Guide decision making within the enterprise

2 key domains inform the development and utilization of architecture

* Enterprise principles: inform how organization sets about fulfilling its mission
* Architecture principles: relate to architecture work, governs the architecture process, defines the rules for use and deploy IT resources

Components of Architecture Principles:

Name (represent the rule’s essence) / Statement (communicate the fundamental rule) / Rationale (business benefits) / Implications (highlight the requirements for carry out the principles)

Developing Architecture Principles:

Typically developed by EA and approved by Board

Informed by principles at Enterprise level, if they exist

Influenced by Enterprise mission and plans, strategic initiatives, external constraints, current systems, industry trends

Qualities of Principles:

Understandable, Robust, Complete, Consistent, Stable

Applying Architecture Principles

* Provide a framework which enterprise can make conscious decisions about EA
* Guide to establish evaluation criteria,
* Defining the functional requirements of architecture

Sometimes principles are related, sometimes they compete. Principles might be self-evident but should be documented.

#### 23.6.1 Business Principles

##### Principle 1: Primacy 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.

##### Principle 2: Maximize Benefit to the Enterprise

Statement:

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

Rationale:

##### 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

##### 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;

##### 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.

##### 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.

##### 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.

##### 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.

##### Principle 9: Protection of Intellectual Property

#### 23.6.2 Data Principles

##### Principle 10: Data is an Asset

Statement:

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

##### 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.

##### 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.

##### Principle 13: Data Trustee

Statement:

Each data element has a trustee accountable for data quality.

##### 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.

##### 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.

#### 

#### 23.6.3 Application Principles

##### Principle 16: Technology Independence

Statement:

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

##### 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.

#### 23.6.4 Technology Principles

##### Principle 18: Requirements-Based Change

Statement:

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

##### Principle 19: Responsive Change Management

Statement:

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

##### 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.

##### 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.

**Stakeholder Management**

Benefits:

Identify most powerful stakeholders early to shape the architecture

Support from most powerful stakeholders help resource

Architect ensure they fully understand the architecture process and benefits of EA

Identify conflicting or competing objectives among stakeholders

Approach:

Use during phase A and updated throughout phase

Identify stakeholders: Who gains/loses from change? Who controls? Who designs?

Consider both the visible (associated with project) and invisible (make contribution but not associated)

Classify Stakeholder Positions: Ask is person ready to change and move towards Target Architecture? Has the person made a commitment to development of EA?

An **Architecture Pattern** expresses a fundamental structural organization or schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them.

A **Design Pattern** provides a scheme for refining the subsystems or components of a software system, or the relationships between them. It describes a commonly recurring structure of communicating components that solves a general design problem within a particular context.

An **Idiom** is a low-level pattern specific to a programming language. An idiom describes how to implement particular aspects of components or the relationships between them using the features of the given language.

**Business Scenarios**

Business scenario describes

1. A business process, application or set of applications enabled by the architecture
2. Business and technology environment,
3. People and the computing components
4. Desired outcome of proper execution

A business scenario represents a business need, and enables vendor to understand the value of developed solution. Used principally the Architecture Vision and the Business Architecture,

Objective: Specific, Measure, Actionable, Realistic, Timebound (SMART)

Benefits: Architecture is based on complete set of requirements, Clear business value, Clear relevant of potential solution, communication with vendors

Approach:

1. Observe, structure information, uncover business rules, stay focus
2. Identify, document and rank the problem
   1. If the problem is too specific or a ‘how’, or too vague or not actionable, raise a red flag
3. Identify the business and technical environment, and document in models
4. Identify the actors, documenting objective

**Gap Analysis**

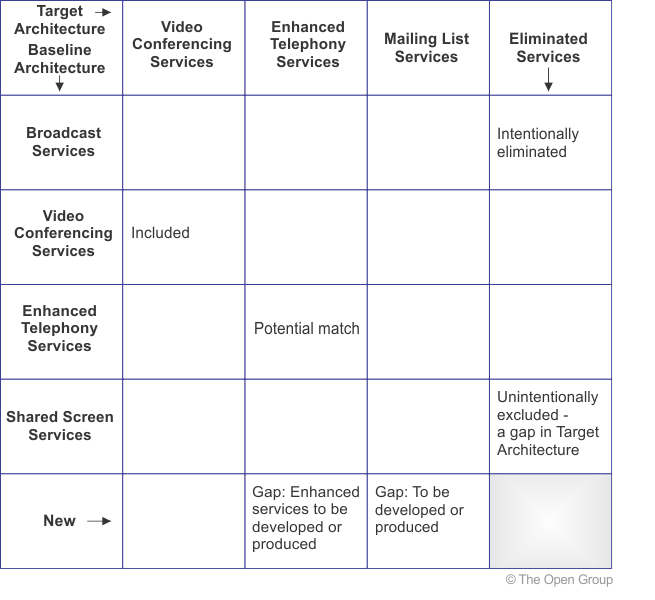
Business domain gaps: people gaps, process gaps, tools gaps, Data domain gaps: data not sufficient

Application impacted, Technologies impacted

Draw a matrix with ABBs of BA on vertical axis, ABBs of TA on horizontal axis

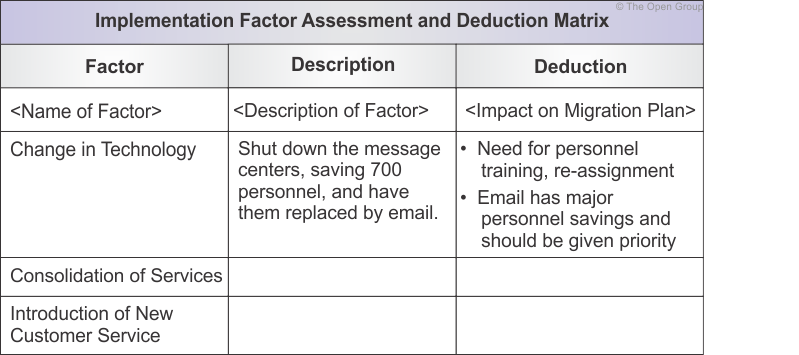
Add a BA axis final label ‘New’, and TA axis final label ‘Eliminated’

* When ABB is in both BA and TA, record ‘Included’
* When ABB in BA is missing in TA, review.
  + If it was correctly eliminated, mark ‘Intentionally Eliminated’ in column ‘Eliminated’
  + If it was not, mark as ‘gap’ in column ‘Eliminated’
* When ABB in TA is missing in BA, mark as ‘Gap’ on ‘New’ row.

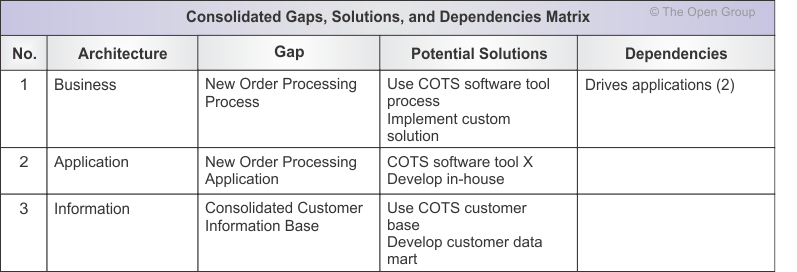
[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/27_gap_eg.png)

**Migration Planning**

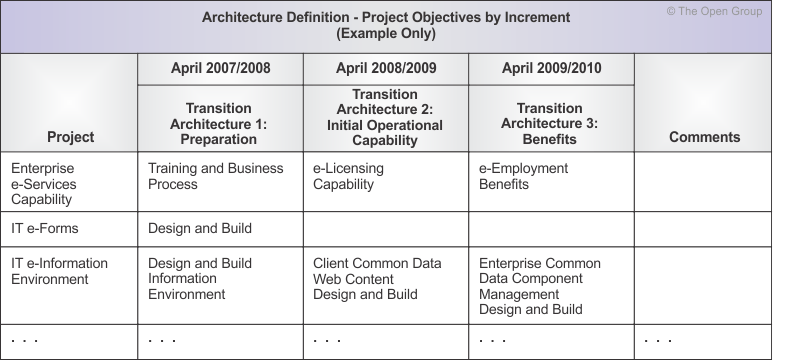
* Create an Implementation Factor Assessment and Deduction matrix to document factors impacting Architecture Implementation and Migration Plan: including
  + factor,
  + descriptions of factor
  + deductions (impact on migration plan)

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/28_matrix_factor.png)

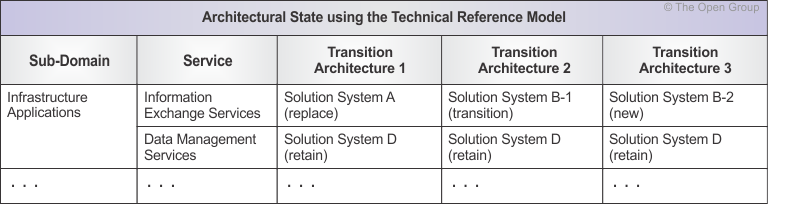
* Use Consolidated Gaps, Solutions and Dependencies matrix to group the identified gaps
  + Assess potential solutions and dependencies to 1 or more gaps

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/28_matrix_consolidated.png)

* Use Architecture Definition Increments Table to plan for a series of Transition Architectures
  + List projects and assign their incremental deliverables across the Transition Architectures

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/28_matrix_increments.png)

* Use a Transition Architecture State Evolution Table
  + Show the proposed state of architectures and various levels using TRM
  + All Solution Building Blocks (SBB) should be described wrt their delivery and impact on these services
  + Use‘retain’, ‘replace’ and ‘ transition’ to indicate target capability

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/28_matrix_trm.png)

* Use Business Value Assessment Technique to evaluate projects
  + Draw up a matrix based on Value Index Dimension / Risk Index Dimension

**Interoperability Requirements**

Phase A: security considerations of information and service exchanges are first reveal

Phase B: information and service exchanges are further defined in business terms

Phase C (Data): content of information exchanges are detailed using corporate data

Phase C (App): the way various app share information and services are specified

Phase D: technical mechanisms to permit information and service exchanges are specified

Phase E: actual solutions are selected

Phase F: interoperability is logically implemented

3 categories:

* Operational
* Information
* Technical

5 IT categories:

* Presentation: common look-and-feel
* Information: corporate information is seamlessly shared to achieve a common set of client information
* Application: corporate functionality is integrated and share-able so that applications are not duplicated
* Technical: common methods / services for communication/storage/data access, based on IT platforms

Enterprise Operating Model

* What is the necessary level of business process for delivering to customers
* May have more than1 type of operating model

Refining Interoperability

* Meets the needs in an unambiguous way
* Upon completion, update the consolidated gap analysis results and dependencies
* Refined interoperability measures (types, targets) should be embedded in the Target Architecture definition and implemented in the Transition Architectures

Determining Interoperability Requirements

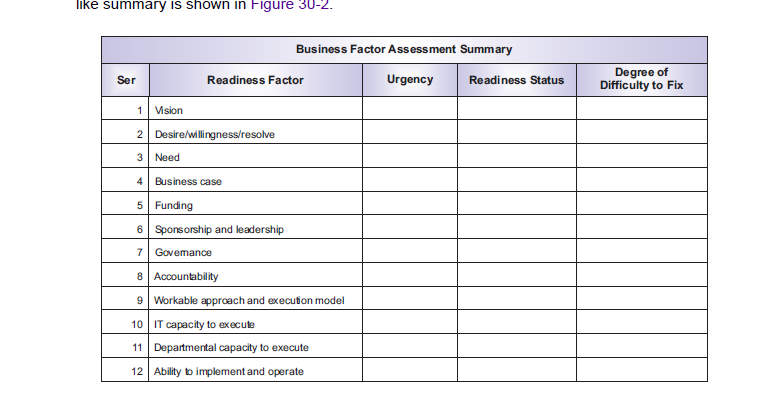
* A matrix showing interoperability requirements is a useful tool
* Start in Phase B to capture the nature of information sharing, and evolve to determine what systems share what information in Phase C.

**Business Transformation Readiness Assessment**

Understanding the readiness of the organization to accept change, identifying the issues, and

then dealing with them in the Implementation and Migration Plans is key to successful

architecture transfor mation in Phases E and F.

****

**Risk Management**

Initial level of risk: prior to determine / implement mitigating actions

Residual level of risk: after implement mitigation actions

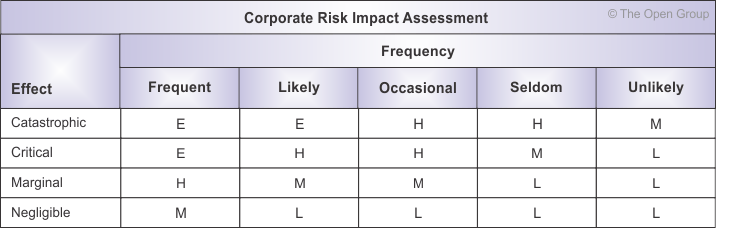
Risk classification

* Normally classify as time/cost/scope/contract risks/corporate risks
* Classify risk by architectural domains (BDAT)

Risk Identification

* Identify the base-line / target states and then identify the actions required to move target state. The implications for not achieving the target state can result in the discovery of risks

Initial Risk Assessment

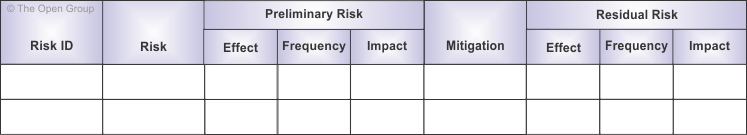


Risk Mitigation and Residual Risk Assessment

* From simple monitoring and/or acceptance of risk, to
* A contingency plan for complete redundancy in a Business Continuity Plan

Conduct Residual Risk Assessment

* Re-assess the effect & frequency then recalculate the impacts



Risk monitoring and governance

**Capability Planning**

* Capability-based planning focuses on planning, engineering, and delivery of strategic business capabilities to the enterprise. Business-driven and business led.
* Frames all phase of architecture development in business outcomes context., links IT vision, architectures (ABB) and Implementation and Migration Plans with corporate business plans
* Ensures business plan drives the enterprise from a top-down approach, while adaptable to leverage bottom-up innovations.

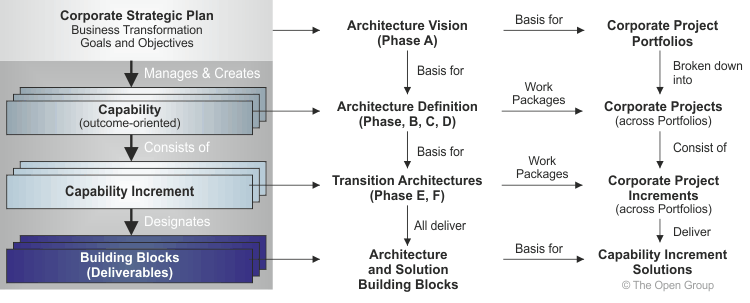
Project led with line-of-business (vertical) does not have corporate (horizontal) perspective. Corporation must cope with delivery of business capabilities with co-ordination across business verticals.

Horizontal: Capability Management (Support cross functional corporate)

Vertical: Functional Management (Support vertical line of business),

Capability dimension:

* Capabilities are engineered to take consideration of various dimensions of corporate portfolio
* People/Process/Material dimension. Each organization has different but similar dimension

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/32_relationships.png)

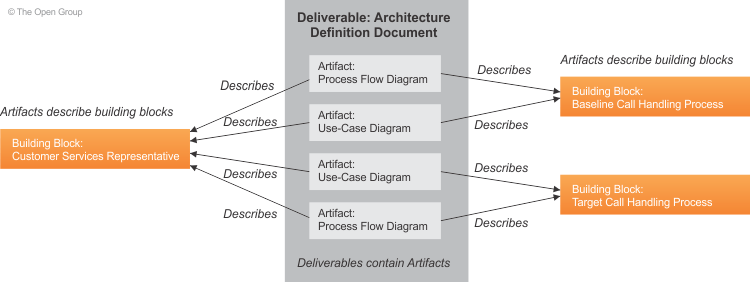
**Architecture Content Framework**

Provides a structural model for architecture content that allows major work products to be consistently defined, structured and presented

Deliverable: project outputs contractually specified and reviewed, archived in Architecture Repository as reference model

Artifact: an aspect of the architecture, specified as catalogs (list of things), matrices (relationship between things) and diagrams (pictures of things). Ex. Use-case diagram. Artifacts form content of Architecture Repository

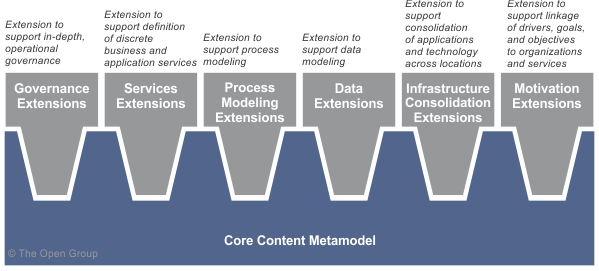
Building block: a component, combine to deliver architectures and solutions ABB describe capability required and shapes SBB. SBB represents components that will be used to implement required capability.

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/02_concepts2.png)

**Content Metamodel:** Definition of a set of entities that allow architecture concepts to be captured and represented, structures architecture information in a way to process to meet stakeholders needs

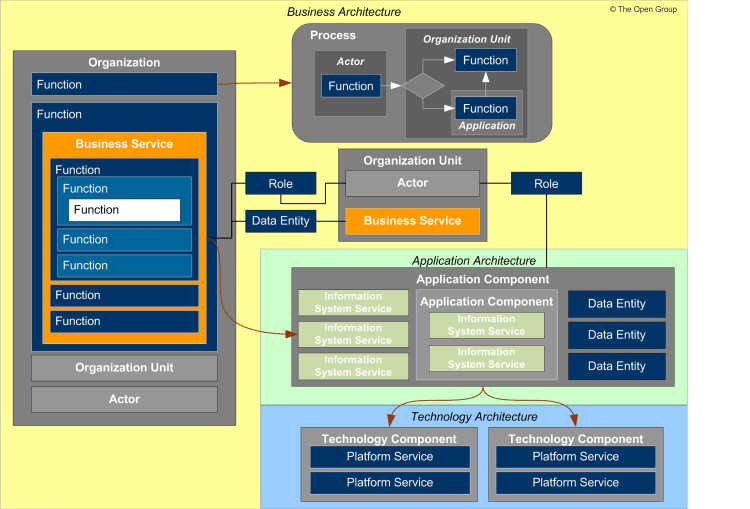
Core content metamodel concepts:

* Core and Extension Content: a basic model with minimum feature set and then support inclusion of optional extensions
* Core metamodel provides a minimum set of architectural content to support traceability across artifacts
* Governance/Services/Process Modeling/Data/Infrastructure Consolidation/Motivation Extensions are optional and are selected in Preliminary Phase. Extensions are suggestions and can be further customized

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/34_contentfwk1.png)

Core metamodel entities

* Actor, Application Component, Business Services, Data Entity, Function, Information System Service, organization unit, Platform Service, Role, Technology Component
  + Process: used to describe flow of interactions between services and functions
  + Functions: units of business capability
  + Business Services: support organizational objectives and defined at a level consistent with level of governance, are deployed onto application components
  + Application components: deployed onto technology components

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/34_contentfwk2.png)

Catalog, Matrix and Diagram Concept

Ad-hoc catalogs, matrices, and diagram can be generated from on-demand querying of Architecture Repository

|  |  |
| --- | --- |
| **ADM Phase** | **Artifacts** |
| Preliminary | Principles Catalog |
| Architecture Vision | Stakeholder Map Matrix |
|  | Value Chain Diagram |
|  | Solution Concept Diagram |
| Business Architecture | Organization/Actor Catalog |
|  | Role Catalog |
|  | Business Service/Function Catalog |
|  | Business Interaction Matrix |
|  | Actor/Role Matrix |
|  | Business Footprint Diagram |
|  | Business Service/Information Diagram |
|  | Functional Decomposition Diagram |
|  | Product Lifecycle Diagram |
| Information Systems | Data Entity/Data Component Catalog |
| (Data Architecture) | Data Entity/Business Function Matrix |
|  | Application/Data Matrix |
|  | Conceptual Data Diagram |
|  | Logical Data Diagram |
|  | Data Dissemination Diagram |
| Information Systems | Application Portfolio Catalog |
| (Application Architecture) | Interface Catalog |
|  | Application/Organization Matrix |
|  | Role/Application Matrix |
|  | Application/Function Matrix |
|  | Application Interaction Matrix |
|  | Application Communication Diagram |
|  | Application and User Location Diagram |
|  | Application Use-Case Diagram |
| Technology Architecture | Technology Standards Catalog |
|  | Technology Portfolio Catalog |
|  | Application/Technology Matrix |
|  | Environments and Locations Diagram |
|  | Platform Decomposition Diagram |
| Opportunities and Solutions | Project Context Diagram |
|  | Benefits Diagram |
| Requirements Management | Requirements Catalog |

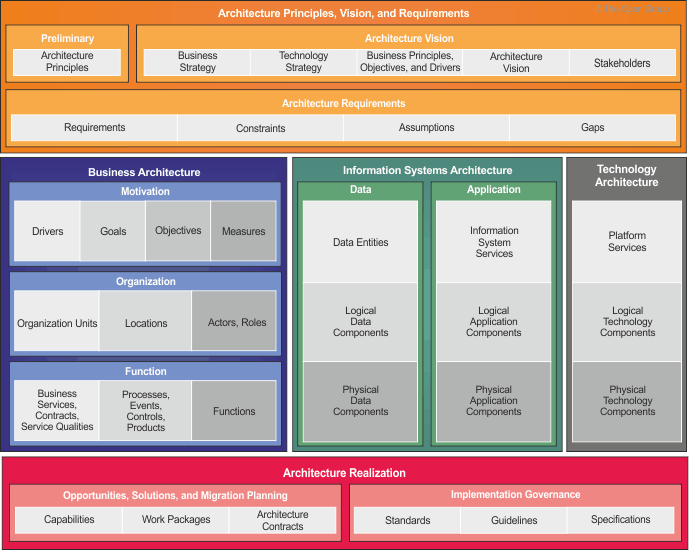
**Content Framework**

provides an underlying structure for the ADM, (i.e. what the architecture should look like)

defines inputs and outputs in more details,

High level Content Framework contains

* Architecture Principles, Vision, and Requirements artifacts: capture context of architecture models, typically collected in Preliminary and Architecture Vision phases
* Business Architecture artifacts: capture architecture models of business operation
* Information System Architecture artifacts: looking at applications and data in line with TOGAF ADM phases
* Technology Architecture artifacts: capture procured technology assets
* Architecture Realization artifacts: capture change roadmaps showing transition between states

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/34_contentfwk5.png)

Content Metamodel in Detail

Core Content Metamodel: describes the metamodel entities that form the core content metamodel

Core Architecture Artifacts: set of artifacts intended to accompany the core content metamodel

|  |  |
| --- | --- |
| Preliminary | Principles Catalog |
| Architecture Vision | Stakeholder Map Matrix |
|  | Value Chain Diagram |
|  | Solution Concept Diagram |
| Business Architecture | Organization/Actor Catalog |
|  | Role Catalog |
|  | Business Service/Function Catalog |

Full Content Metamodel: metamodel entities that form extensions to content metamodel When all extensions are applied to the core content metamodel, a number of new metamodel entities are introduced

Governance Extension: allow additional structured data to be held against objectives and services

* When an organization is considering IT change that will result in a significant impact to

existing operational governance models

* When an organization is looking to transform its operational governance practice
* Service levels are defined in a more structured way

Services extension: allow more sophisticated modeling of the service portfolioby creating a concept of IS services in addition to the core concept of business services.

* When the business has a preset definition of its services that does not align well to technical and architectural needs
* When business and IT use different language to describe similar capabilities

Process modeling extension: allow detailed modeling of process flows by adding events, products, and controls to the metamodel.

* Where the architecture must pay specific attention to state and events
* Where the architecture is required to explicitly identify and store process control steps

Data extension: allow more sophisticated modeling and the encapsulation of data. The core model provides a data entity concept which supports the creation of data models

* Where the architecture features significant complexity and risk around the location, encapsulation, and management of or access to data
* The structure of data is modeled independently from its location, allowing data models to be developed that span multiple systems without being tied to physical concerns.

Infrastructure consolidation extension: in landscapes where the application and technology portfolios have become fragmented and the architecture seeks to consolidate the business

* Where many technology products are in place with duplicate or overlapping capability
* Where many applications are in place with duplicate or overlapping functionality

Motivation extension: allow additional structured modeling of the drivers, goals,and objectives that influence an organization to provide business services to its customers.

* When the architecture needs to understand the motivation of organizations in more detail

than the standard business or engagement principles

**Architectural Artifacts**

Concerns:

* key interests that are crucially important to the stakeholders in the 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.

View:

* a representation of a whole system from the perspective of a related set of concerns. Comprise of selected parts of one or more models to show to stakeholders their concerns are being addressed in the design of the system architecture.
* Enable the architecture to be communicated to and understood by the stakeholders, so they can verify that the system will address their concerns.

Viewpoints:

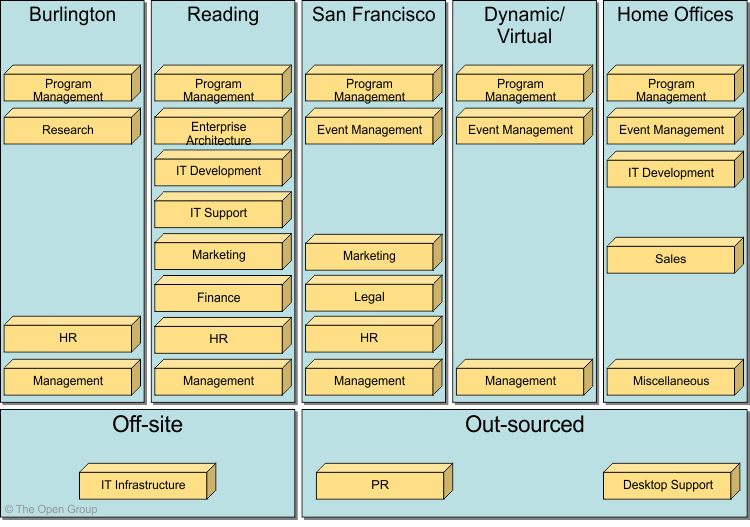
* defines how to construct and use a view (by means of an appropriate schema or template); the information that should appear in the view; the modeling techniques; and a rationale for these choices

A view is what you see. A viewpoint is where you are looking from - the vantage point or perspective that determines what you see.

Viewpoints are generic, and can be stored in libraries for re-use. A view is always specific to the architecture for which it is created.

The viewpoint is specified as follows:

|  |  |
| --- | --- |
| **Viewpoint Element** | **Description** |
| Stakeholders | Management Board, Chief Executive Officer |
| Concerns | Show the top-level relationships between geographical sites and business functions. |
| Modeling technique | Nested boxes diagram. Outer boxes = locations; inner boxes = business functions. Semantics of nesting = functions performed in the locations. |



Developing Views in ADM:

Steps:

1. Refer to an existing library of viewpoints
2. Select the appropriate viewpoints (based on the stakeholders and concerns that need to be covered by views)
3. Generate views of the system by using the selected viewpoints as templates

This approach can be expected to bring the following benefits:

* Less work for the architects (because the viewpoints have already been defined)
* Better comprehensibility for stakeholders (because the viewpoints are already familiar)
* Greater confidence in the validity of the views (because their viewpoints have a known track record)

The architect may choose to develop a new viewpoint that will cover the outstanding need, and then generate a view from it, or an *ad hoc* view for a specific system and later consider whether a generalized form of the implicit viewpoint should be defined explicitly and saved in a library.

The architect should be aware that every view has a viewpoint, at least implicitly, and that defining the viewpoint in a systematic way (as recommended by ISO/IEC 42010:2007) will help in assessing its effectiveness; i.e., does the viewpoint cover the relevant stakeholder concerns?

The following describes catalogs, matr ices, and diagrams that may be created within Phase A

(Architecture Vision) as listed in Section 7.5.

**Phase A**

**Stakeholder Map Matrix**

The purpose of the Stakeholder Map matrix is to identify the stakeholders for the architecture

engagement, their influence over the engagement,

**Phase B**

**Organization/Actor Catalog**

The purpose of the Organization/Actor catalog is to capture a definitive listing of all participants

that interact with IT, including users and owners of IT systems.

The Organization/Actor catalog can be referenced when developing

**Driver/Goal/Objective Catalog**

The purpose of the Driver/Goal/Objective catalog is to provide a cross-organizational reference

of how an organization meets

**Business Service/Function Catalog**

The purpose of the Business Service/Function catalog is to provide a functional decomposition

in a for m that can be filtered, reported on, and queried, as a supplement to graphical Functional

Decomposition diagrams.

**Location Catalog**

The Location catalog provides a listing of all locations where an enterpr ise carr ies out business

operations or houses architecturally relevant assets, such as data centers or end-user

computing equipment.

**Phase C**

**Data Entity/Data Component Catalog**

The purpose of the Data Entity/Data Component catalog is to identify and maintain a list of all

the data use across the enterprise

The Data Entity/Data Component catalog contains the following metamodel entities:

n Data Entity

n Logical Data Component

n Physical Data Component

**Data Entity/Business Function Matrix**

The purpose of the Data Entity/Business Function matrix is to depict the relationship between

data entities and business functions within the enterpr ise. Business functions are supported by

business services with explicitly defined boundaries and will be supported and realized by

business processes. The mapping of the Data Entity-Business Function relationship enables the

following to take place:

n Assign ownership of data entities to organizations

n Understand the data and infor mation exchange requirements business services

**Application/Data Matrix**

The purpose of the Application/Data matrix is to depict the relationship between applications

(i.e., application components) and the data entities that are accessed and updated by them.

**Data Dissemination Diagram**

The purpose of the Data Dissemination diagram is to show the relationship between data entity,

business service, and application components. The diagram shows how the logical entities are

to be physically realized by application components. This allows effective sizing to be carried out

and the IT footprint to be refined.

**Phase C**

**Application/Function Matrix**

The purpose of the Application/Function matrix is to depict the relationship between applications

and business functions within the enterprise.

Business functions are performed by organizational units.

Understand the application support requirements of the business services and processes

carr ied out

n Suppor t the gap analysis and determine whether any of the applications are missing and

as a result need to be created

**Application Communication Diagram**

The purpose of the Application Communication diagram is to depict all models and mappings

related to communication between applications in the metamodel entity.

It shows application components and interfaces between components.

**Application Interaction Matrix**

The purpose of the Application Interaction matrix is to depict communications relationships

between applications.

**Application and User Location Diagram**

The Application and User Location diagram shows the geographical distribution of applications.

It can be used to show where applications are used by the end user;

The diagram enables:

n Identification of the number of package instances needed to sufficiently support the user

population that may be spread out geographically

n Estimation of the number and the type of user licenses for the package or other software

**Phase D**

**Technology Portfolio Catalog**

The purpose of this catalog is to identify and maintain a list of all the technology in use across

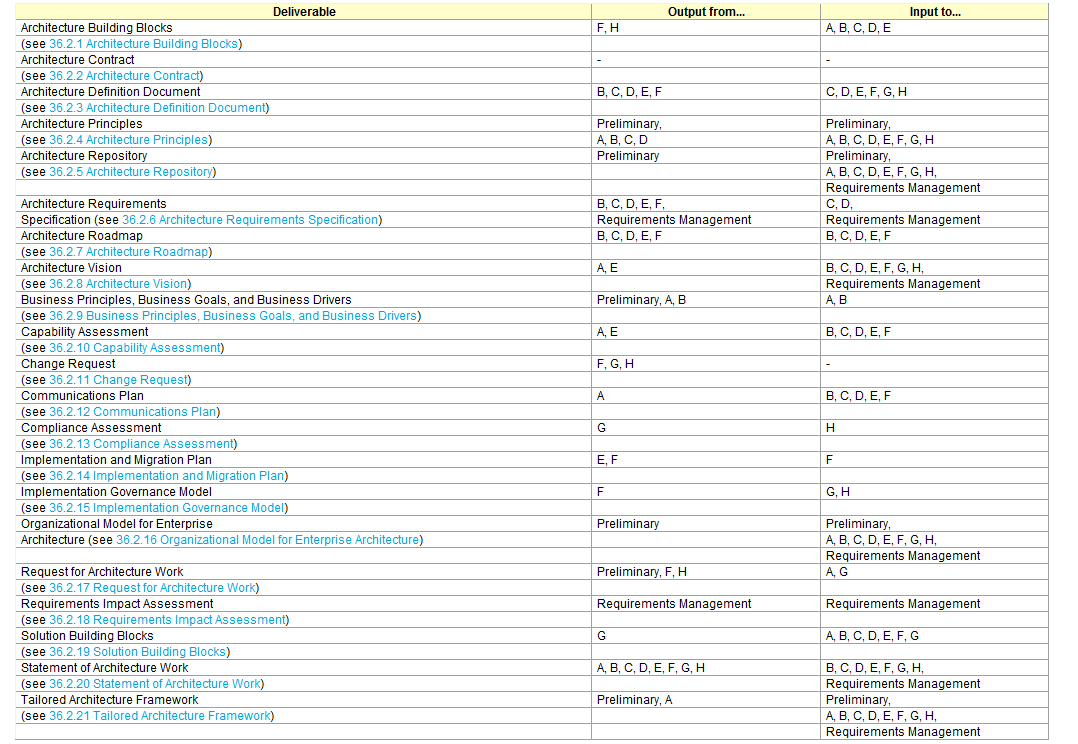
the enterprise,

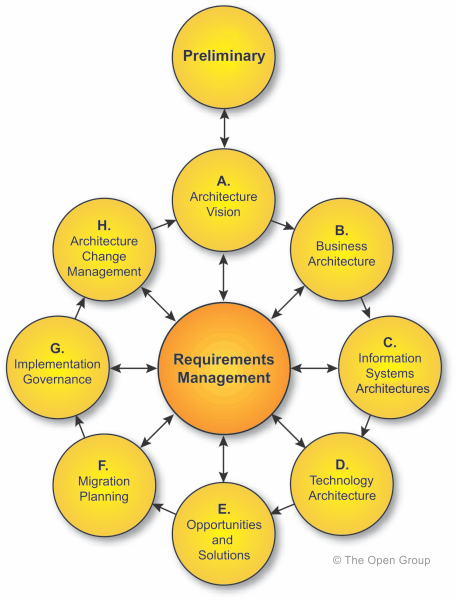
**Application/Technology Matrix**

The Application/Technology matrix documents the mapping of applications to technology

platform.

**Architecture Artifacts**





* **ABB**: Output from Migration Planning / Architecture Change Management
* **Architecture Contract**: Not an ADM input / output
* **ADD**: Contains scope, baseline / transition architecture, gap analysis, mapping to architecture repository. Output from B-F, input for all except A/B.
* **Architecture Principles:** general rules and guidelines, output of preliminary phase, A-D, input to all phases
* **Architecture Repository:** holding area for all architecture-related projects, to locate re-usable assets. Output from Preliminary phase, input to Preliminary, all phases, requirements mgt
* **Architecture Requirements Specification:** Quantitative statements to outline what a project must do to comply with architecture. Output from B-F, Requirements Mgt, Input to C,D, Requirements Mgt
* **Architecture Roadmap:**  lists individual work packages to realize Target Architecture and lays them on a timeline. Contains transition architectures, consolidated gaps, solutions and dependencies matrix, SBBs, Risk and Issues. Output from B-F, Input to B-F
* **Architecture Vision:** summary of changes, provide stakeholders with a formally agreed outcome. Output from Architecture Vision (A) and Opportunities and Solutions (E), Input to B-H, Requirement Mgt
* **Business Principles, Goals, Drivers**
* **Capability Assessment:** Business Capability / IT Capability / Architecture maturity/ BTRA
* **Change Request**
* **Communication Plans**
* **Compliance Assessment**
* **Implementation and Migration Plan:** Output from E/F to F
* **Implementation Governance Model:** Output from F to G and H
* **Organizational Model for EA:** output Preliminary to all phases
* **Request for Architecture Work:** output from Preliminary phase, Migration Plan (F) Architecture Change Management (H), input to Architecture Vision (A) and Implementation Governance (G)
* **Requirements Impact Assessment**
* **Solution Building Blocks:** Output from Implementation Governance (G), input to all
* **Statement of Architecture Work**
* **Tailored Architecture Framework**

**Architecture Deliverables**

As deliverables are typically the contractual or formal work products of an architecture project,

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.

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.

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

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

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.

Implementation and Migration Plan

The Implementation and Migration Plan provides a schedule of the projects that will realize the Target Architecture.

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.

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

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

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.

Typical contents of an Organizational Model for enterprise architecture are:

* Scope of organizations impacted
* Maturity assessment, gaps, and resolution approach
* Roles and responsibilities for architecture team(s)

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.

**Building Blocks**

* Package of functionality to meet business needs
* Has a type that corresponds to TOGAF content metamodel (actor, business service, application or data entity)
* Has a defined boundary
* Interoperate with other inter-dependent building blocks

Characteristics

* Considers implementation and usage, and evolves to exploit standard, assembled from other building blocks, re-usable, replaceable and well specified.

Building block’s boundary and specification should be loosely coupled to its implementation

* Can realize building block in several different ways
* Good choice of BBs can improve legacy system integration, interoperability and flexibility of new system creation
* Level of defined details various depend on what stage of architecture development

Architecture BB

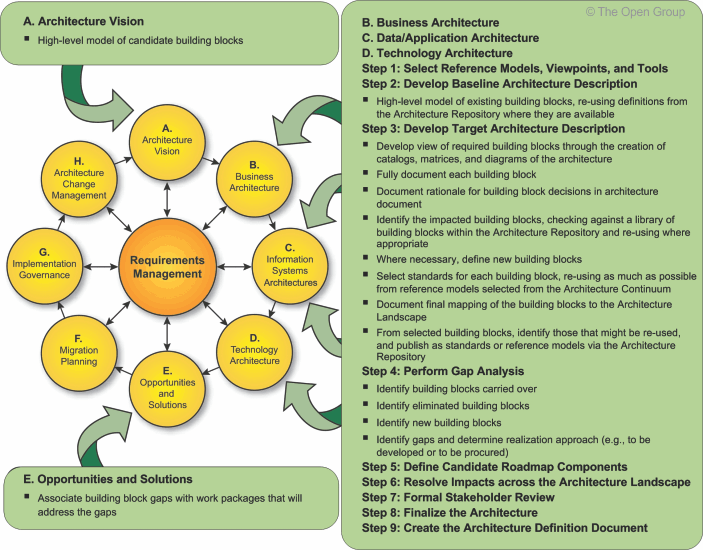
* ABBs capture architecture requirements BDAT
* Direct and guide the development of SBBs
* Includes
  + Fundamental functionality and attributes: semantic, unambiguous
  + Interfaces: chosen or supplied set
  + Interoperability and relationship with other building blocks
  + Map to business entities and policies

Solution BB (may either be procured or developed)

* Define what products and components will implement the functionality
* Define the implementation
* Fulfill business requirements
* Product and vendor-aware
* Includes:
  + Specific functionality and attributes
  + Interface: the implemented set
  + Required SBBs used with required functionality and names of interfaces
  + Mapping from SBB to IT topology and operational policies
  + Specification of attributes shared across environment (i.e. scalability, security)
  + Design driver and constraints
  + Relationships between SBBs and ABBs

Building blocks in Architecture Design:

* Architecture need only contain building blocks that are relevant to business problem
* Building blocks have complex relationship to one another
* Building blocks should conform to standards relevant to their type
* Consider 3 classes of building blocks
  + Re-usable building blocks
  + Building blocks to the subject of development
  + Building blocks to the subject of purchase

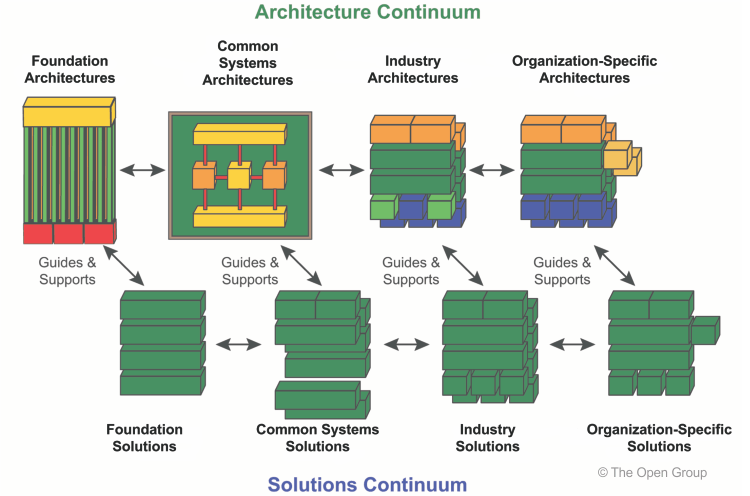


**Enterprise Continuum**

3 distinct continuum

* Enterprise: outermost classifies assets related to the context of the overall EA
* Architecture: a restructuring of ABBs, shows relationships among TOGAF, common system architectures (III-RM), industry architectures and EA. Discover common and eliminate redundancy
* Solution: what is available as re-usable SBB, address commonality and differences among the product

Architecture Continuum

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/39_entcon.png)

A Foundation Architecture consists of generic components, inter-relationships, principles, and

guidelines that provide a foundation on which more specific architectures can be built

Common Systems Architectures guide the selection and integration of specific services from the

Foundation Architecture to create an architecture useful to create an architecture useful for building common (i.e., highly reusable) solutions across a wide number of relevant domains.

Industr y Architectures guide the integration of common systems components with industry specific

components, and guide the creation of industry solutions for targeted customer problems within a particular industry. A typical example of an industry-specific component is a data model representing the business

Functions

Organization-Specific Architectures describe and guide the final deployment of solution

components for a particular enterpr ise Defines building blocks specific to a particular enterpr ise. Contains organization-specific business models, data, applications, and technologies

**Architecture Partitioning**

Partition because

Organizational unit architectures conflict with 1 another

Allows specific groups of architects to own and develop specific elements

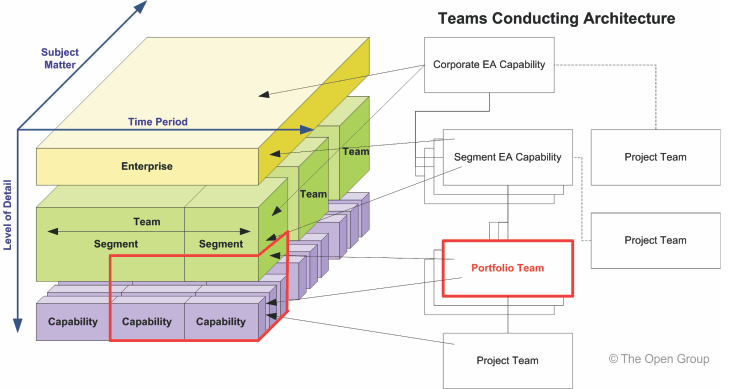
Effective architecture re-use

Each enterprise need to adopt a partitioning model that reflects its own model

Steps within Preliminary Phase

1. Determine organization structure: governance bodies, team memberships, team reporting
2. Determine responsibilities for each standing team (Subject matter areas, level of details, time period, stakeholders)
3. Determine relationships between architectures

Team allocation

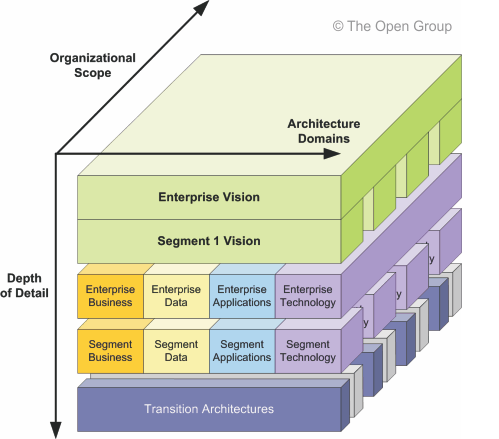
[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/40_partitioning5.png)

Integration

Integration across the architectural domains provides a cross-domain view of the state of a segment of the enterprise for a point in time.

* Integration across the organizational scope of the business provides a cross-segment view of the enterprise.
* The Architecture Vision provides an integrated summary of Architecture Definitions, which provide an integrated summary of Transition Architectures.

In order to mitigate against this risk, standards for content integration should be defined and architecture governance should address content integration as a condition of architectural compliance. For example, a standard catalog of business processes can be agreed for an enterprise

[](http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/40_partitioning7.png)

**Architecture Repository**

At a high level, six classes of architectural information are expected to be held within an Architecture Repository:

* The **Architecture Metamodel** describes the organizationally tailored application of an architecture framework, including a method for architecture development and a metamodel for architecture content.
* The **Architecture Capability** defines the parameters, structures, and processes that support governance of the Architecture Repository.
* The **Architecture Landscape** presents an architectural representation of assets in use, or planned, by the enterprise at particular points in time.
* The **Standards Information Base** captures the standards with which new architectures must comply, which may include industry standards, selected products and services from suppliers, or shared services already deployed within the organization.
* The **Reference Library** provides guidelines, templates, patterns, and other forms of reference material that can be leveraged in order to accelerate the creation of new architectures for the enterprise.
* The **Governance Log** provides a record of governance activity across the enterprise.

**Architecture Landscape:** holds architectural views of the state of the enterprise at particular points in time. Due to the sheer volume and the diverse stakeholder needs throughout an entire enterprise, the Architecture Landscape is divided into three levels of granularity:

1. **Strategic Architectures** show a long-term summary view of the entire enterprise. Strategic Architectures provide an organizing framework for operational and change activity and allow for direction setting at an executive level.
2. **Segment Architectures** provide more detailed operating models for areas within an enterprise. Segment Architectures can be used at the program or portfolio level to organize and operationally align more detailed change activity.
3. **Capability Architectures** show in a more detailed fashion how the enterprise can support a particular unit of capability. Capability Architectures are used to provide an overview of current capability, target capability, and capability increments and allow for individual work packages and projects to be grouped within managed portfolios and programs.

**Reference Library:** hold reference materials that should be used to develop architectures

The Reference Library should contain:

* Reference Architectures
* Reference Models
* Viewpoint Library
* Templates

The terms *reference architecture* and *reference model* are not used carefully in most literature. Reference architecture and reference model have the same relationship as architecture and model. Either can exist as either generic or an organization-specific state

**Standards Information Base:**

At the top level, standards are classified in line with the TOGAF architecture domains, including the following areas:

* **Business Standards**:
  + Standard shared business functions
  + Standard role and actor definitions
  + Security and governance standards for business activity
* **Data Standards**:
  + Standard coding and values for data
  + Standard structures and formats for data
  + Standards for origin and ownership of data
  + Restrictions on replication and access
* **Applications Standards**:
  + Standard/shared applications supporting specific business functions
  + Standards for application communication and interoperation
  + Standards for access, presentation, and style
* **Technology Standards**;
  + Standard hardware products
  + Standard software products
  + Standards for software development

The Governance Log should contain the following items:

* **Decision Log**: A log of all architecturally significant decisions that have been made in the organization.
* **Compliance Assessments**
* **Capability Assessments**:
* **Calendar**:
* **Project Portfolio**:
* **Performance Measurement**:

**The Enterprise Repository:** While the Architecture Repository holds information concerning the enterprise architecture and associated artifacts there are a considerable number of enterprise repositories that support the architecture. These include the Requirements Repository storing requirements and the Solutions Repository storing Solution Building Blocks

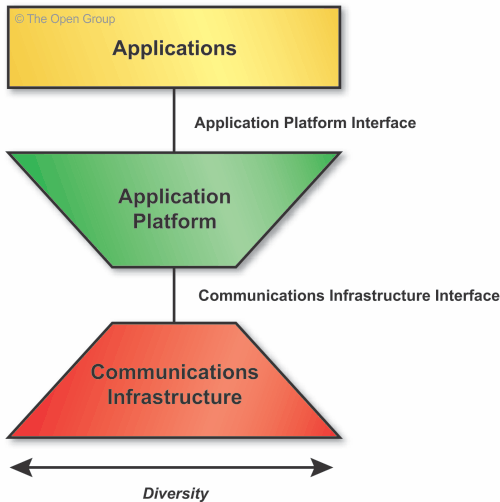
**Foundation Architecture: Technical Reference Model**

Foundation Architecture is an architecture of generic services that specific architectures and components can be built. It is embodied with TRM, which provides a model and taxonomy of generic platform services.

2 components 1) Taxonomy: terminology and coherent description of components, 2) TRM graphic: visual representation

Iit is important to emphasize that the use of TOGAF, and in particular the TOGAF ADM, is in no way dependent on use of the TOGAF TRM taxonomy. Other taxonomies are perfectly possible, and may be preferable for some organizations.

Breakdown



The high-level TRM seeks to emphasize two major common architectural objectives:

1. **Application Portability**, via the Application Platform Interface - identifying the set of services that are to be made available in a standard way to applications via the platform
2. **Interoperability**, via the Communications Infrastructure Interface - identifying the set of Communications Infrastructure services that are to be leveraged in a standard way by the platform

2 categories of Application Software:

* **Business Applications**, which implement business processes for a particular enterprise or vertical industry. (i.e. Patient record management services used in the Medical industry)
* **Infrastructure Applications**, which provide general-purpose business functionality, based on infrastructure services. (i.e. Workflow services)

During development of the Technology Architecture, business applications and infrastructure applications are important sources of requirements for Technology Architecture services

**Application Platform**

It is important to recognize that the Application Platform in the TOGAF TRM is a single, generic, conceptual entity. From the viewpoint of the TOGAF TRM, the Application Platform contains all possible services. In a specific Target Architecture, the Application Platform will contain only those services needed to support the required functions.

Moreover, the Application Platform for a specific Target Architecture will typically not be a single entity, but rather a combination of different entities for different, commonly required functions, such as desktop client, file server, print server, application server, Internet server, database server, etc.,

The set of services identified and defined for the Application Platform will change over time. New services will be required as new technology appears and as application needs change

**Communication Infrastructure:**

The Communications Infrastructure provides the basic services to interconnect systems and provide the basic mechanisms for opaque transfer of data. It contains the hardware and software elements which make up the networking and physical communications links used by a system, and of course all the other systems connected to the network.

**III Reference Model**

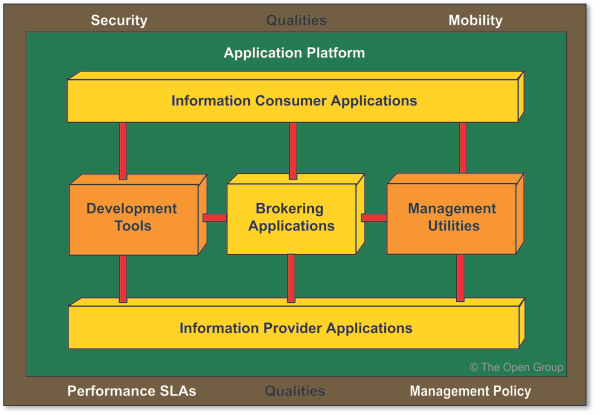
Like the TOGAF TRM, the III-RM has two main components:

1. A **taxonomy**, which defines terminology, and provides a coherent description of the components and conceptual structure of an integrated information infrastructure
2. An associated **III-RM graphic**, which provides a visual representation of the taxonomy, and the inter-relationship of the components, as an aid to understanding

Boundary-less Information Flow: getting information to the right people at the right time in a secure, reliable manner, in order to support the operations that are core to the extended enterprise

To gain significant operational efficiencies and improve the many different business processes of the enterprise - both internal processes, and those spanning the key interactions with suppliers, customers, and partners - if only I could provide my staff with:

* **Integrated information** so that different and potentially conflicting pieces of information are not distributed throughout different systems
* **Integrated access to that information** so that staff can access all the information they need and have a right to, through one convenient interface



* **Business Applications**, denoted by the yellow boxes in the high-level model (corresponding to the "Business Applications" box in the TRM graphic). There are three types of Business Application in the model:
  + **Brokering Applications**, which manage the requests from any number of clients to and across any number of Information Provider Applications
  + **Information Provider Applications**, which provide responses to client requests and rudimentary access to data managed by a particular server
  + **Information Consumer Applications**, which deliver content to the user of the system, and provide services to request access to information in the system on the user's behalf
* **Infrastructure Applications**, denoted by the orange boxes in the high-level model (corresponding to the "Infrastructure Applications" box in the TRM graphic). There are two types of Infrastructure Application in the model:
  + **Development Tools**, which provide all the necessary modeling, design, and construction capabilities to develop and deploy applications that require access to the integrated information infrastructure, in a manner consistent with the standards of the environment
  + **Management Utilities**, which provide all the necessary utilities to understand, operate, tune, and manage the run-time system in order to meet the demands of an ever-changing business, in a manner consistent with the standards of the environment
* An **Application Platform**, which provides supporting services to all the above applications - in areas such as location, directory, workflow, data management, data interchange, etc. - and thereby provides the ability to locate, access, and move information within the environment. This set of services constitutes a subset of the total set of services of the TRM Application Platform, and is denoted by the dark green underlay in the high-level model (corresponding to the Application Platform in the TRM graphic).
* The **Interfaces** used between the components. Interfaces include formats and protocols, application programming interfaces, switches, data values, etc. Interfaces among components at the application level are colored red. Interfaces between any application-level components and their supporting services in the Application Platform are colored white (corresponding to the API box in the TRM graphic).
* The **Qualities** backplane, denoted by the brown underlay in the high-level model (corresponding to the Qualities backplane in the TRM graphic). The Application Software and Application Platform must adhere to the policies and requirements depicted by the qualities backplane.

**Architecture Capabililty**

In order to successfully operate an architecture function within an enterpr ise, it is necessar y to

put in place appropriate organization structures, processes, roles, responsibilities, and skills to

realize the Architecture Capability.

Part VII: Architecture Capability Framework provides a set of reference materials for how to

establish such an architecture function

**Architecture Board**

The Architecture Board is typically made responsible, and accountable, for achieving some or all of the following goals:

* Providing the basis for all decision-making with regard to the architectures
* Consistency between sub-architectures
* Establishing targets for re-use of components
* Flexibility of enterprise architecture:
  + To meet changing business needs
  + To leverage new technologies
* Enforcement of Architecture Compliance
* Improving the maturity level of architecture discipline within the organization
* monitoring and control of the Architecture Contract
* Ensuring compliance with the architectures, and granting dispensations that are in keeping with the technology strategy and objectives

From a governance perspective, the Architecture Board is also responsible for:

* The production of usable governance material and activities
* Providing a mechanism for the formal acceptance and approval of architecture through consensus and authorized publication

Operation of the Architecture Board

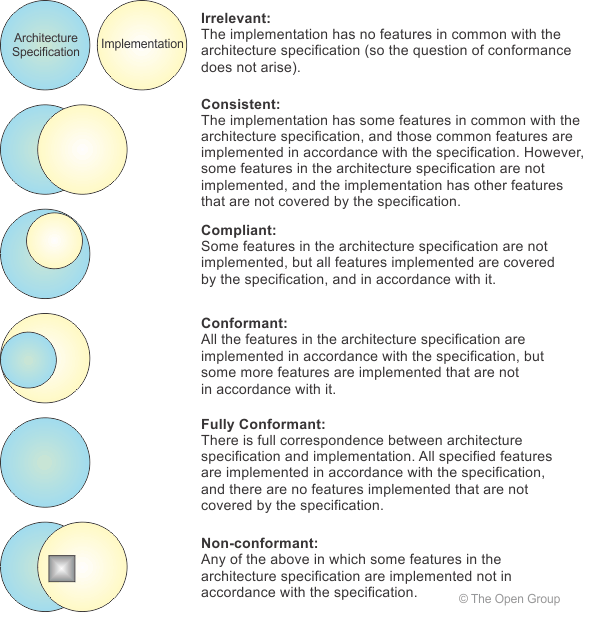
These meetings will provide key direction in:

* Supporting the production of quality governance material and activities
* Providing a mechanism for formal acceptance through consensus and authorized publication
* Providing a fundamental control mechanism for ensuring the effective implementation of the architectures

**Architecture Compliance**

IT governance function within an enterprise will normally define two complementary processes:

* The **Architecture** function will be required to prepare a series of Project Architectures; i.e., project-specific views of the enterprise architecture that illustrate how the enterprise architecture impacts on the major projects within the organization. (See ADM Phases A to F.)
* The **IT Governance** function will define a formal Architecture Compliance review process (see [48.3 Architecture Compliance Reviews](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap48.html#tag_48_03)) for reviewing the compliance of projects to the enterprise architecture.



The goals of an Architecture Compliance review include some or all of the following:

* First and foremost, catch errors in the project architecture early, and thereby reduce the cost and risk of changes required later in the lifecycle
* The Architecture Compliance review can be a good way of deciding between architectural alternatives, since the business decision-makers typically involved in the review can guide decisions in terms of what is best for the business,

Compliance reviews are held at appropriate project milestones or checkpoints in the project's lifecycle. Specific checkpoints should be included as follows:

* Development of the architecture itself (ADM compliance)
* Implementation of the architecture(s) (architecture compliance)

The aim is to hold the review as soon as practical, at a stage when there is still time to correct any major errors or shortcomings

The main roles in the process are tabulated below.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Role** | **Responsibilities** | **Notes** |
| 1 | Architecture Board | To ensure that IT architectures are consistent and support overall business needs. | Sponsor and monitor architecture activities. |
| 2 | Project Leader (or Project Board) | Responsible for the whole project. |  |
| 3 | Architecture Review Co-ordinator | To administer the whole architecture development and review process. | More likely to be business-oriented than technology-oriented. |
| 4 | Lead Enterprise Architect | To ensure that the architecture is technically coherent and future-proof. | An IT architecture specialist. |
| 5 | Architect | One of the Lead Enterprise Architect's technical assistants. |  |
| 6 | Customer | To ensure that business requirements are clearly expressed and understood. | Manages that part of the organization that will depend on the success of the IT described in the architecture. |
| 7 | Business Domain Expert | To ensure that the processes to satisfy the business requirements are justified and understood. | Knows how the business domain operates; may also be the customer. |
| 8 | Project Principals | To ensure that the architects have a sufficiently detailed understanding of the customer department's processes. They can provide input to the business domain expert or to the architects. | Members of the customer's organization who have input to the business requirements that the architecture is to address. |

 Main steps in the process are tabulated below.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Action** | **Notes** | **Who** |
| 1 | Request architecture review | As mandated by IT governance policies and procedures. | Anyone, whether IT or business-oriented, with an interest in or responsibility for the business area affected. |
| 2 | Identify responsible part of organization and relevant project principals. |  | Architecture Review Co-ordinator |
| 3 | Identify Lead Enterprise Architect and other architects. |  | Architecture Review Co-ordinator |
| 4 | Determine scope of review | Identify which other business units/departments are involved. Understand where the system fits in the corporate architecture framework. | Architecture Review Co-ordinator |
| 5 | Tailor checklists. | To address the business requirements. | Lead Enterprise Architect |
| 6 | Schedule Architecture Review Meeting |  | Architecture Review Co-ordinator with collaboration of Lead Enterprise Architect. |
| 7 | Interview project principals | To get background and technical information:   * For internal project: in person * For COTS: in person or via RFP   Use checklists. | Lead Enterprise Architect and/or Architect, Project Leader, and Customers |
| 8 | Analyze completed checklists | Review against corporate standards. Identify and resolve issues. Determine recommendations. | Lead Enterprise Architect |
| 9 | Prepare Architecture Compliance review report | May involve supporting staff. | Lead Enterprise Architect |
| 10 | Present review findings | To Customer To Architecture Board | Lead Enterprise Architect |
| 11 | Accept review and sign off |  | Architecture Board and Customer |
| 12 | Send assessment report/summary to Architecture Review Co-ordinator |  | Lead Enterprise Architect |

**Architecture Contracts**

Architecture Contracts may occur at various stages of the Architecture Development Method (ADM); for example:

* The Statement of Architecture Work created in Phase A of [Part II: Architecture Development Method (ADM)](http://pubs.opengroup.org/architecture/togaf9-doc/arch/pt2.html) is effectively an Architecture Contract
* The development of one or more architecture domains (business, data, application, technology), and in some cases the oversight of the overall enterprise architecture, may be contracted out to systems integrators, applications providers, and/or service providers.
* At the beginning of Phase G (Implementation Governance), between the architecture function and the function responsible for implementing the enterprise architecture defined in the preceding ADM phases.
* When the enterprise architecture has been implemented (at the end of Phase G),

Content

The Statement of Architecture Work is created as a deliverable of Phase A, and is effectively an Architecture Contract between the architecting organization and the sponsor of the enterprise architecture

Between the Architecture Design and Development Partners

Typical contents of an Architecture Design and Development Contract are:

* Introduction and background
* The nature of the agreement
* Scope of the architecture
* Architecture and strategic principles and requirements

Between the Architecting Function and Business Users

Typical contents of a Business Users' Architecture Contract are:

* Scope
* Strategic requirements
* Architecture deliverables that meet the business requirements
* Architecture business metrics
* Service architecture (includes Service Level Agreement (SLA))

**Architecture Governance**

Corporate governance: Cor porate governance is thus a broad topic, beyond the scope of an enterpr ise architecture

* framework such as TOGAF.

Technology governance: Technology governance controls how an organization utilizes technology in the research,

IT governance IT governance provides the framework and structure that links IT resources and infor mation to reach enter prise goals and strategies

Architecture governance: Architecture governance is the practice and orientation by which enterpr ise architectures and

other architectures are managed and controlled at an enterpr ise-wide level. It includes the

Characteristics

Architecture governance is the practice and orientation by which enterprise architectures and other architectures are managed and controlled at an enterprise-wide level. It includes the following:

* Implementing a system of controls over the creation and monitoring of all architectural components and activities, to ensure the effective introduction, implementation, and evolution of architectures within the organization
* Implementing a system to ensure compliance with internal and external standards and regulatory obligations
* Establishing processes that support effective management of the above processes within agreed parameters
* Developing practices that ensure accountability to a clearly identified stakeholder community, both inside and outside the organization

Key Success factors

* Best practices for the submission, adoption, re-use, reporting, and retirement of architecture policies, procedures, roles, skills, organizational structures, and support services
* Organizational responsibilities and structures to support the architecture governance processes and reporting requirements
* Integration of tools and processes to facilitate the take-up of the processes, both procedurally and culturally
* Criteria for the control of the architecture governance processes, dispensations, compliance assessments, SLAs, and OLAs
* Internal and external requirements for the effectiveness, efficiency, confidentiality, integrity, availability, compliance, and reliability of all architecture governance-related information, services, and processes

There are three important elements of architecture governance strategy that relate particularly to the acceptance and success of architecture within the enterprise. While relevant and applicable in their own right apart from their role in governance, and therefore described separately, they also from an integral part of any effective architecture governance strategy.

* A cross-organizational Architecture Board (see [*47. Architecture Board*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap47.html#tag_47)) must be established with the backing of top management to oversee the implementation of the IT governance strategy.
* A comprehensive set of architecture principles (see [*23. Architecture Principles*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap23.html#tag_23)) should be established, to guide, inform, and support the way in which an organization sets about fulfilling its mission through the use of IT.
* An Architecture Compliance (see [*48. Architecture Compliance*](http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap48.html#tag_48)) strategy should be adopted - specific measures (more than just a statement of policy) to ensure compliance with the architecture, including Project Impact Assessments, a formal Architecture Compliance review process, and possibly including the involvement of the architecture team in product procurement.

**Architecture Maturity Model**

**Level 0: None**

No enterprise architecture program. No enterprise architecture to speak of.

**Level 1: Initial**

Informal enterprise architecture process underway.

1. Processes are *ad hoc* and localized. Some enterprise architecture processes are defined. There is no unified architecture process across technologies or business processes. Success depends on individual efforts.
2. Enterprise architecture processes, documentation, and standards are established by a variety of *ad hoc* means and are localized or informal.

**Level 2: Under Development**

Enterprise architecture process is under development.

1. Basic enterprise architecture process is documented based on OMB Circular A-130 and Department of Commerce Enterprise Architecture Guidance. The architecture process has developed clear roles and responsibilities.
2. IT vision, principles, business linkages, Baseline, and Target Architecture are identified. Architecture standards exist, but not necessarily linked to Target Architecture. Technical Reference Model (TRM) and Standards Profile framework established.
3. Explicit linkage to business strategies.
4. Management awareness of architecture effort.

**Level 3: Defined**

Defined enterprise architecture including detailed written procedures and TRM.

1. The architecture is well defined and communicated to IT staff and business management with operating unit IT responsibilities. The process is largely followed.
2. Gap analysis and Migration Plan are completed. Fully developed TRM and Standards Profile. IT goals and methods are identified.
3. Enterprise architecture is integrated with capital planning and investment control.
4. Senior management team aware of and supportive of the enterprise-wide architecture process. Management actively
5. considered in identifying projects.

**Level 4: Managed**

Managed and measured enterprise architecture process.

1. Enterprise architecture process is part of the culture. Quality metrics associated with the architecture process are captured.
2. Enterprise architecture documentation is updated on a regular cycle to reflect the updated enterprise architecture. Business, Data, Application, and Technology Architectures defined by appropriate *de jure* and *de facto* standards.
3. Capital planning and investment control are adjusted based on the feedback received and lessons learned from updated enterprise architecture. Periodic re-examination of business drivers.
4. Senior management team directly involved in the architecture review process.
5. The entire operating unit accepts and actively participates in the enterprise architecture process.
6. Architecture documents are updated regularly, and frequently reviewed for latest architecture developments/standards.
7. Explicit governance of all IT investments. Formal processes for managing variances feed back into enterprise architecture.
8. All planned IT acquisitions and purchases are guided and governed by the enterprise architecture.

**Level 5: Optimizing**

Continuous improvement of enterprise architecture process.

1. Concerted efforts to optimize and continuously improve architecture process.
2. A standards and waivers process is used to improve architecture development process.
3. Architecture process metrics are used to optimize and drive business linkages. Business involved in the continuous process improvements of enterprise architecture.
4. Senior management involvement in optimizing process improvements in architecture development and governance.
5. Feedback on architecture process from all operating unit elements is used to drive architecture process improvements.
6. improvements.

**Architectural Skills**

EA Role

* **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.
* **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.
* **Validate, refine, and expand the model**: verify assumptions, bring in subject matter experts, etc.
* **Manage the architecture**: continuously monitor the models and update them as necessary to show changes, additions, and alterations.

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.

#### 52.6.3 Key Characteristics of an Enterprise Architect

##### 52.6.3.1 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,

##### 52.6.3.2 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.

##### 52.6.3.3 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.

##### 52.6.3.4 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.

##### 52.6.3.5 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.

##### 52.6.3.6 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,

##### 52.6.3.7 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.