The Open Group Standard

TOGAF® Standard — Applying the ADM

The Open Group

The Open Group Standard

TOGAF® Standard — Applying the ADM

ISBN: 1-947754-90-4
Document Number: C220

Published by The Open Group, 2005-2022.

Any comments relating to the material contained in this document may be submitted by email to:

ogspecs@opengroup.org

Contents

List of Figures

Preface

The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through technology standards. With more than 870 member organizations, we have a diverse membership that spans all sectors of the technology community — customers, systems and solutions suppliers, tool vendors, integrators and consultants, as well as academics and researchers.

The mission of The Open Group is to drive the creation of Boundaryless Information Flow™ achieved by:

- Working with customers to capture, understand, and address current and emerging requirements, establish policies, and share best practices
- Working with suppliers, consortia, and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies
- Offering a comprehensive set of services to enhance the operational efficiency of consortia
- Developing and operating the industry's premier certification service and encouraging procurement of certified products

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/library.

The TOGAF® Standard

The TOGAF Standard is an open, industry consensus framework for Enterprise Architecture.

It is a foundational framework, which means that it is applicable to the development of any kind of architecture in any context. This foundational framework is supplemented by The Open Group TOGAF Library,[1] an extensive and growing portfolio of guidance material, providing practical guidance in the application of the TOGAF framework in specific contexts.

The TOGAF Documentation

The TOGAF documentation consists of a set of documents:

- The TOGAF Standard, which describes the generally applicable approach to Enterprise and IT Architecture
- The TOGAF Library, a portfolio of additional guidance material, which supports the practical application of the TOGAF approach

[1] The TOGAF Library (see www.opengroup.org/togaf-library) is a structured library of resources that support the TOGAF Standard.

This Document

This is the TOGAF Standard — Applying the ADM.

Intended Audience

The TOGAF Standard is intended for Enterprise Architects, Business Architects, IT Architects, Data Architects, Systems Architects, Solution Architects, and anyone responsible for the architecture function within an organization.

Acknowledgements

The Open Group is grateful for the contribution of many individuals and organizations in the development of the TOGAF Standard. See the TOGAF Standard — Introduction and Core Concepts for details.

Trademarks

ArchiMate, DirecNet, Making Standards Work, Open O logo, Open O and Check Certification logo, The Open Group, TOGAF, UNIX, UNIXWARE, and the Open Brand X logo are registered trademarks and Boundaryless Information Flow, Build with Integrity Buy with Confidence, Commercial Aviation Reference Architecture, Dependability Through Assuredness, Digital Practitioner Body of Knowledge, DPBoK, EMMM, FACE, the FACE logo, FHIM Profile Builder, the FHIM logo, FPB, Future Airborne Capability Environment, IT4IT, the IT4IT logo, O-AA, O-DEF, O-HERA, O-PAS, Open Agile Architecture, Open FAIR, Open Footprint, Open Process Automation, Open Subsurface Data Universe, Open Trusted Technology Provider, OSDU, Sensor Integration Simplified, SOSA, and the SOSA logo are trademarks of The Open Group.

The Open Group acknowledges that there may be other company names and products that might be covered by trademark protection and advises the reader to verify them independently.

Referenced Documents

Please refer to the TOGAF Standard — Introduction and Core Concepts: Appendix A for documents referenced in the TOGAF Standard.

Chapter 1: Introduction

This chapter provides an introduction to the guidance provided in the TOGAF Standard — Applying the ADM (this document).

The Architecture Development Method (ADM) process can be adapted to deal with many different usage scenarios, including different process styles (e.g., the use of iteration) and also specific specialist architectures (such as security). Guidelines included within this document are as follows:

- Using the TOGAF Framework with Different Architecture Styles (see Section 1.1) discusses how the framework can be adapted to different architectural styles
- Applying Iteration to the ADM (see Chapter 2) discusses the concept of iteration and shows potential strategies for applying iterative concepts to the ADM
- Applying the ADM across the Architecture Landscape (see Chapter 3) discusses the different types of architecture engagement that may occur at different levels of the enterprise — this section then also discusses how the ADM process can be focused to support different types of engagement
- Architecture Partitioning (see Chapter 4) discusses how partitions are used to simplify the development and management of the Enterprise Architecture

1.1 Using the TOGAF Framework with Different Architecture Styles

The TOGAF framework is designed to be flexible and is used with various architectural styles.

Architectural styles differ in terms of focus, form, techniques, materials, subject, and time period. The TOGAF Standard is a generic framework intended to be used in a wide variety of environments. It is a flexible and extensible framework that can be readily adapted to a number of architectural styles.

An organization's Architecture Landscape can be expected to contain architecture work that is developed in many architectural styles. The TOGAF Standard ensures that the needs of each stakeholder are appropriately addressed in the context of other stakeholders and the Baseline Architecture.

When using the TOGAF Standard to support a specific architectural style the practitioner must take into account the combination of distinctive features in which architecture is performed or expressed. As a first step, the distinctive features of a style must be identified.

The second step is determining how these distinctive features will be addressed. Addressing a distinctive style should not call for significant changes to the TOGAF framework; 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 (see the TOGAF Standard — ADM Techniques). Depending upon the distinctive features, different architectural styles will add new elements that must be described, highlight existing elements, adjust the notation used to describe the architecture, and focus the architect on some stakeholders or stakeholder concerns.

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. Dominance of a particular architectural style can direct the practitioner to revisit the Preliminary Phase to make changes to the Architecture Capability or to address a distinctive feature in the expected scope of a single ADM cycle.

Style-specific reference models and maturity models are commonly used tools that support a practitioner.

During the lifetime of the TOGAF framework many architectural styles have been developed to address key problems facing practitioners and to demonstrate how the TOGAF framework can be made more relevant within defined contexts.

Some of these have been developed by The Open Group Forums and Work Groups working in specific areas and have been published in Guides, White Papers, and Standards. Examples include:

- TOGAF® Series Guide: Using the TOGAF® Framework to Define and Govern Service-Oriented Architectures
- TOGAF® Series Guide: Integrating Risk and Security within a TOGAF® Enterprise Architecture

Some of these have been developed collaboratively between The Open Group and other bodies. Examples include:

- TOGAF® and SABSA® Integration
- Archi Banking Group: Combining the BIAN Reference Model, ArchiMate® Modeling Notation, and the TOGAF® Framework
- Exploring Synergies between TOGAF® and Frameworx
- TOGAF® 9 and DoDAF 2.0

The TOGAF Library (see www.opengroup.org/togaf-library) is a structured library of resources that support the TOGAF Standard.

Chapter 2: Applying Iteration to the ADM

2.1 Overview

The graphical representation of the TOGAF ADM and the description of the ADM phases discretely in order, as shown in the TOGAF Standard — Architecture Development Method, can be read to imply a deterministic waterfall methodology. This method of presentation is provided for the purpose of quickly communicating the basics of architecture development and the architecture development cycle. In practice, two key concepts are used to manage the complexity of developing an Enterprise Architecture and managing its lifecycle — iteration and levels (see Chapter 3). The two concepts are tightly linked.

The ADM supports a number of concepts that are characterized as iteration. First, iteration describes the process of describing a comprehensive Architecture Landscape through multiple ADM cycles based upon individual initiatives bound to the scope of the Request for Architecture Work. Second, iteration describes the integrated process of developing an architecture where the activities described in different ADM phases interact to produce an integrated architecture. In order to concisely describe the activity and outputs, this latter iteration is described in sequential terms. Third, iteration describes the process of managing change to the organization's Architecture Capability.

Iteration to develop a comprehensive Architecture Landscape:

- Projects will exercise through the entire ADM cycle, commencing with Phase A

  Each cycle of the ADM will be bound by a Request for Architecture Work. The architecture output will populate the Architecture Landscape, either extending the landscape described, or changing the landscape where required.

- Separate projects may operate their own ADM cycles concurrently, with relationships between the different projects

- One project may trigger the initiation of another project

  Typically, this is used when higher-level architecture initiatives identify opportunities or solutions that require more detailed architecture, or when a project identifies landscape impacts outside the scope of its Request for Architecture Work.

Iteration within an ADM cycle (Architecture Development iteration):

- Projects may operate multiple ADM phases concurrently

  Typically, this is used to manage the inter-relationship between Business Architecture, Information Systems Architecture, and Technology Architecture.

- Projects may cycle between ADM phases, in planned cycles covering multiple phases

  Typically, this is used to converge on a detailed Target Architecture when higher-level architecture does not exist to provide context and constraint.

- Projects may return to previous phases to update work products with new information

  Typically, this is used to converge on an executable Architecture Roadmap or Implementation and Migration Plan, when the implementation details and scope of change trigger a change or re-prioritization of stakeholder requirements.

Iteration to manage the Architecture Capability (Architecture Capability iteration):

- Projects may require a new iteration of the Preliminary Phase to (re-)establish aspects of the Architecture Capability identified in Phase A to address a Request for Architecture Work
- Projects may require a new iteration of the Preliminary Phase to adjust the organization's Architecture Capability as a result of identifying new or changed requirements for Architecture Capability as a result of a Change Request in Phase H

2.2 Iteration Cycles

The suggested iteration cycles for the TOGAF ADM are shown in Figure 2-1, and can be used to effectively group related architectural activities to achieve a specific purpose. These iteration cycles are referenced in Section 2.3 and Section 2.5.

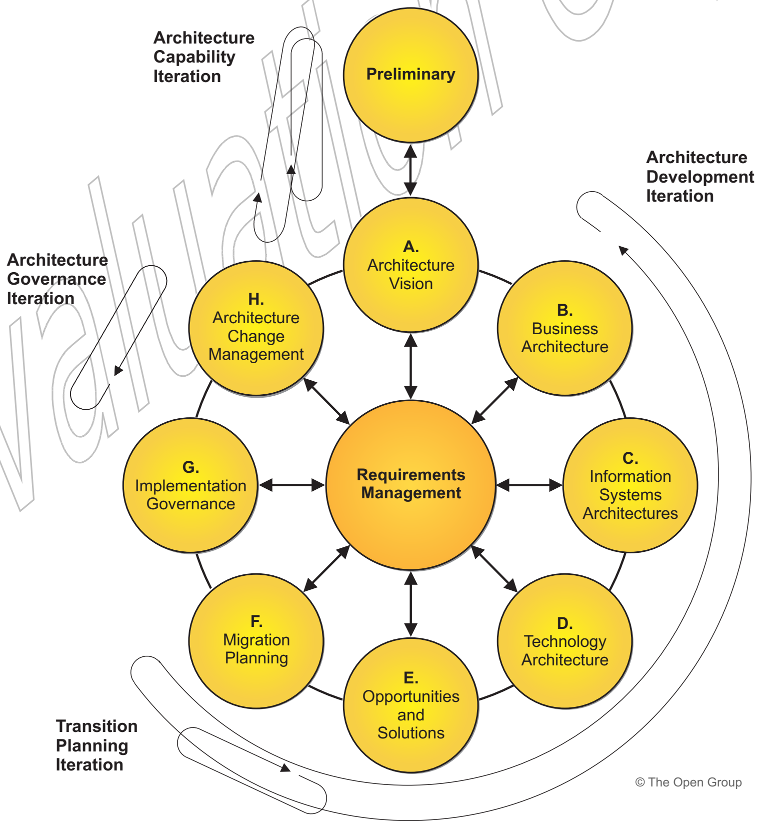

Architecture Capability Iteration
Preliminary
Architecture Development Iteration
Architecture Governance Iteration
A. Architecture Vision
H. Architecture Change Management
B. Business Architecture
G. Implementation Governance
Requirements Management
C. Information Systems Architectures
F. Migration Planning
D. Technology Architecture
E. Opportunities and Solutions
Transition Planning Iteration
© The Open Group

Figure 2-1 Iteration Cycles

- **Architecture Capability** iterations support the creation[1] and evolution of the required Architecture Capability

  This includes the initial mobilization of the architecture activity for a given purpose or architecture engagement type by establishing or adjusting the architecture approach, principles, scope, vision, and governance.

- **Architecture Development** iterations allow the creation of architecture content by cycling through, or integrating, Business, Information Systems, and Technology Architecture phases

  These iterations ensure that the architecture is considered as a whole. In this type of iteration stakeholder reviews are typically broader. As the iterations converge on a target, extensions into the Opportunities & Solutions and Migration Planning phases ensure that the architecture's implementability is considered as the architecture is finalized.

- **Transition Planning** iterations support the creation of formal change roadmaps for a defined architecture

- **Architecture Governance** iterations support governance of change activity progressing towards a defined Target Architecture

2.3 Classes of Architecture Engagement

An architecture function or services organization may be called upon to assist an enterprise in a number of different contexts, as the architectures developed can range from summary to detail, broad to narrow coverage, and current state to future state. In these contexts the concept of iteration should be used in developing the architecture.

Typically, there are three areas of engagement for architects:

- **Identification of Required Change**: outside the context of any change initiative, architecture can be used as a technique to provide visibility of the IT capability in order to support strategic decision-making and alignment of execution

- **Definition of Change**: where a need to change has been identified, architecture can be used as a technique to define the nature and extent of change in a structured fashion

  Within large-scale change initiatives, architectures can be developed to provide detailed Architecture Definition for change initiatives that are bounded by the scope of a program or portfolio.

- **Implementation of Change**: architecture at all levels of the enterprise can be used as a technique to provide design governance to change initiatives by providing big-picture visibility, supplying structural constraints, and defining criteria on which to evaluate technical decisions

Figure 2-2 and the following table show the classes of Enterprise Architecture engagement.

[1] Guidance on how to use a full ADM cycle for initially establishing an organization's Architecture Capability is found in the TOGAF Standard — Enterprise Architecture Capability and Governance.

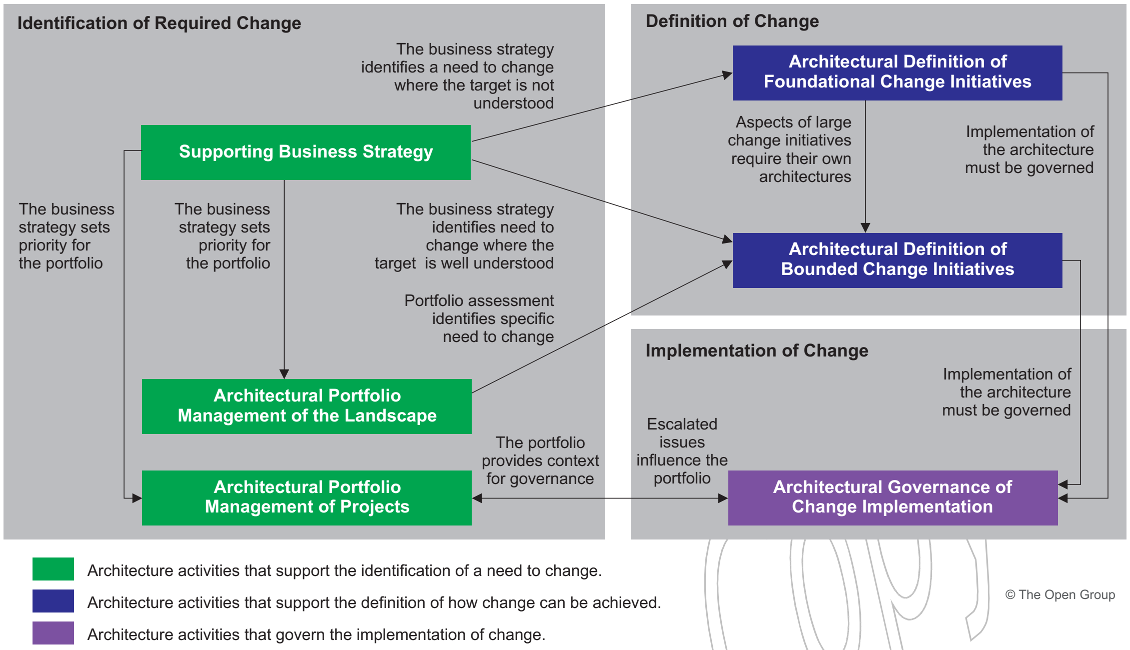

Identification of Required Change
The business strategy identifies a need to change where the target is not understood
Supporting Business Strategy
The business strategy sets priority for the portfolio
The business strategy sets priority for the portfolio
The business strategy identifies need to change where the target is well understood
Portfolio assessment identifies specific need to change
Architectural Portfolio Management of the Landscape
The portfolio provides context for governance
Architectural Portfolio Management of Projects
Definition of Change
Architectural Definition of Foundational Change Initiatives
Aspects of large change initiatives require their own architectures
Implementation of the architecture must be governed
Architectural Definition of Bounded Change Initiatives
Implementation of Change
Implementation of the architecture must be governed
Escalated issues influence the portfolio
Architectural Governance of Change Implementation
🟩 Architecture activities that support the identification of a need to change.
🟦 Architecture activities that support the definition of how change can be achieved.
🟪 Architecture activities that govern the implementation of change.
© The Open Group

Figure 2-2 Classes of Enterprise Architecture Engagement

Each of these architecture engagement types is described in the table below.

Area of Engagement	Architecture Engagement	Description
Identification of Required Change	Supporting Business Strategy	As the business strategies, objectives, goals, and drivers change, it is necessary for the enterprise to change in order to maintain alignment. The creation of new business strategies can be supported by Enterprise Architecture by: ■ Providing visibility of change opportunities ■ Providing elaboration on the practical impacts of a particular strategic choice ■ Providing tests on the feasibility or viability of a particular strategic direction

Area of Engagement	Architecture Engagement	Description
	Architectural Portfolio Management of the Landscape	It is common practice across large organizations for a service management organization to provide operational reporting and management of the IT portfolio. Enterprise Architecture can add a further dimension to service management reporting, by supporting a linkage between operational performance and the strategic need for IT. Using the traceability between IT and business inherent in Enterprise Architecture, it is possible to evaluate the IT portfolio against operational performance data and business needs (e.g., cost, functionality, availability, responsiveness) to determine areas where misalignment is occurring and change needs to take place.
	Architectural Portfolio Management of Projects	It is common practice across large organizations for a program management organization to provide operational reporting and management of the change portfolio. Enterprise Architecture can add a further dimension to project portfolio management reporting, by supporting a linkage between project scope, architectural impact, and business value. Architectural factors can be added to other quantitative project factors to support strategic decision-making on project priority and funding levels.
Definition of Change	Architectural Definition of Foundational Change Initiatives	Foundational change initiatives are change efforts that have a known objective, but are not strictly scoped or bounded by a shared vision or requirements. In foundational change initiatives, the initial priority is to understand the nature of the problem and to bring structure to the definition of the problem. Once the problem is more effectively understood, it is possible to define appropriate solutions and to align stakeholders around a common vision and purpose.

Area of Engagement	Architecture Engagement	Description
	Architectural Definition of Bounded Change Initiatives	Bounded change initiatives are change efforts that typically arise as the outcome of a prior architectural strategy, evaluation, or vision. In bounded change initiatives, the desired outcome is already understood and agreed upon. The focus of architectural effort in this class of engagement is to effectively elaborate a baseline solution that addresses the identified requirements, issues, drivers, and constraints.
Implementation of Change	Architectural Governance of Change Implementation	Once an architectural solution model has been defined, it provides a basis for design and implementation. In order to ensure that the objectives and value of the defined architecture are appropriately realized, it is necessary for continuing Architecture Governance of the implementation process to support design review, architecture refinement, and issue escalation.

Different classes of architecture engagement at different levels of the enterprise will require focus in specific areas, as shown below.

Engagement Type	Focus Iteration Cycles	Scope Focus
Supporting Business Strategy	Architecture Capability Architecture Development (Baseline First)	Broad, shallow consideration given to the Architecture Landscape in order to address a specific strategic question and define terms for more detailed architecture efforts to address strategy realization.
Architectural Portfolio Management of the Landscape	Architecture Capability Architecture Development (Baseline First)	Focus on physical assessment of baseline applications and technology infrastructure to identify improvement opportunities, typically within the constraints of maintaining business as usual.
Architectural Portfolio Management of Projects	Transition Planning Architecture Governance	Focus on projects, project dependencies, and landscape impacts to align project sequencing in a way that is architecturally optimized.

Engagement Type	Focus Iteration Cycles	Scope Focus
Architectural Definition of Foundational Change Initiatives	Architecture Capability Architecture Development (Baseline First) Transition Planning	Focus on elaborating a vision through definition of baseline and identifying what needs to change to transition the baseline to the target.
Architectural Definition of Bounded Change Initiatives	Architecture Development (Target First) Transition Planning	Focus on elaborating the target to meet a previously defined and agreed vision, scope, or set of constraints. Use the target as a basis for analysis to avoid perpetuation of baseline, sub-optimal architectures.
Architectural Governance of Change Implementation	Architecture Governance	Use the Architecture Vision, constraints, principles, requirements, Target Architecture definition, and transition roadmap to ensure that projects realize their intended benefit, are aligned with each other, and are aligned with wider business need.

2.4 Approaches to Architecture Development

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.

In practical terms, an architecture team will always give informal consideration to the baseline when analyzing the target (and *vice versa*). In situations where baseline and target are expected to be considered in parallel by stakeholders, it is recommended that the architecture team focuses priority on one state in order to maintain focus and consistency of execution.

2.5 Iteration Considerations

Some iteration cycles can be executed once, whereas others have a natural minimum number of cycles. For some iteration cycles, each iteration follows the same process; where there is more than one iteration within a cycle, the process differs slightly for each of the iterations.

When considering the usage of iteration cycles, it is also necessary to consider where to place appropriate checkpoints within the process. If the expected level of stakeholder involvement is high, it may be sensible to carry out very frequent but informal checkpoints to ensure that the process is moving in the intended direction. If stakeholders are less closely involved, then checkpoints may be less frequent but more formal. Checkpoints at the completion of each iteration cycle, or at the end of several iteration cycles, are common.

2.5.1 Iteration between ADM Cycles

Each iteration completes an ADM cycle at a single level of Architecture Description. This approach to the ADM uses Phase F (Migration Planning) to initiate new more detailed architecture development projects. This approach is illustrated in Figure 2-3. This type of iteration highlights the need for higher-level architecture to guide and constrain more detailed architecture. It also highlights that the complete Architecture Landscape is developed by multiple ADM iterations.

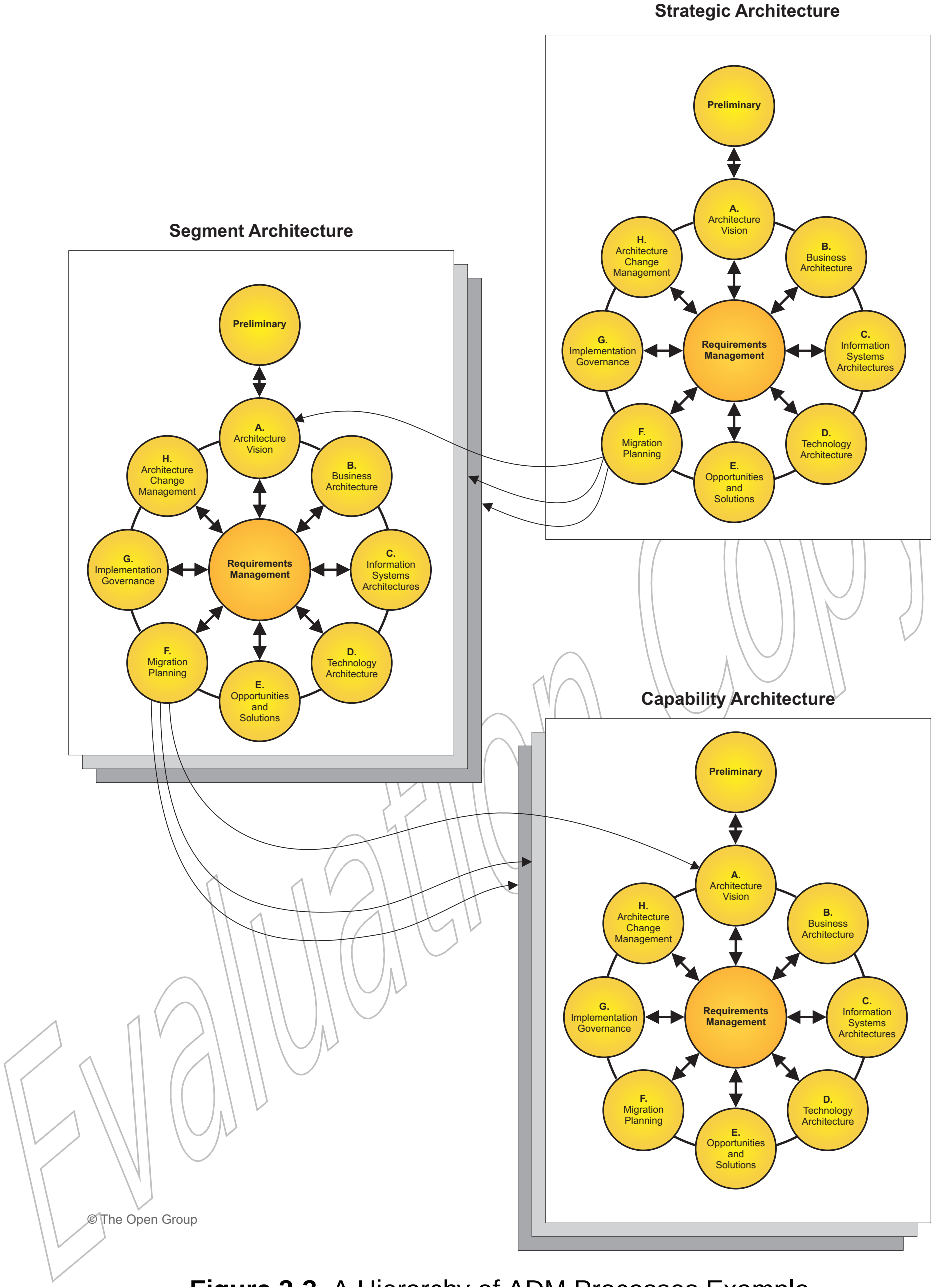
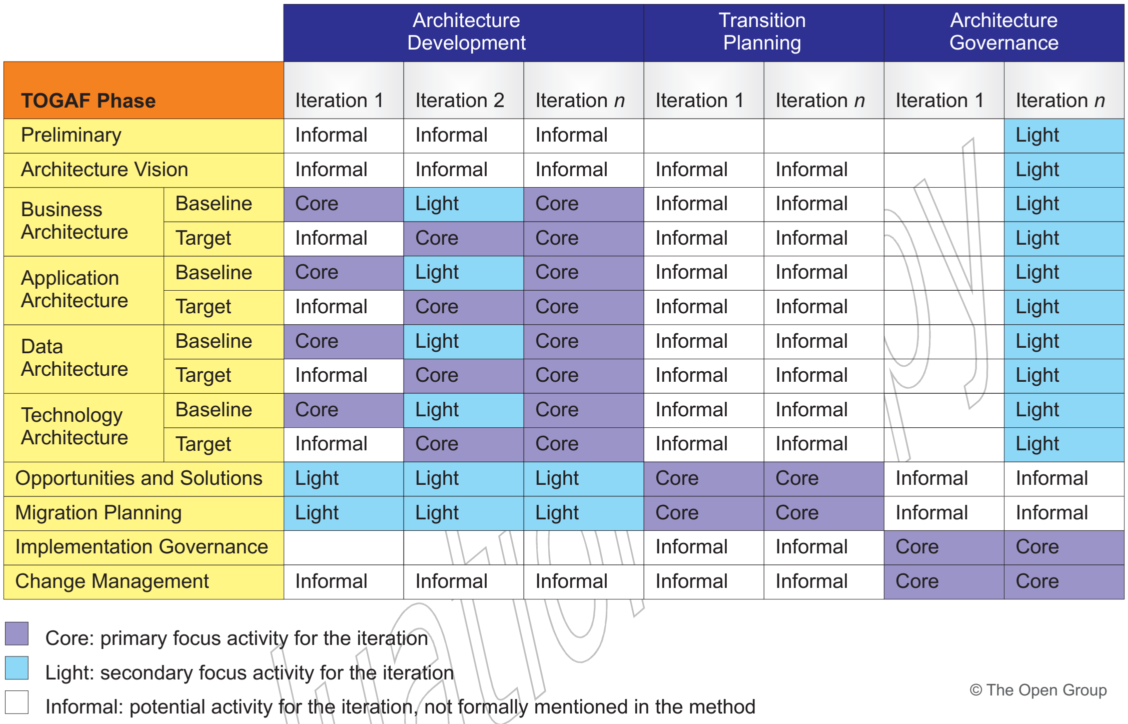


Figure 2-3 A Hierarchy of ADM Processes Example

2.5.2 Iteration within an ADM Cycle

Each iteration cycle crosses multiple TOGAF ADM phases. 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).



		Architecture Development			Transition Planning		Architecture Governance	
TOGAF Phase		Iteration 1	Iteration 2	Iteration *n*	Iteration 1	Iteration *n*	Iteration 1	Iteration *n*
Preliminary		Informal	Informal	Informal				Light
Architecture Vision		Informal	Informal	Informal	Informal	Informal		Light
Business Architecture	Baseline	Core	Light	Core	Informal	Informal		Light
	Target	Informal	Core	Core	Informal	Informal		Light
Application Architecture	Baseline	Core	Light	Core	Informal	Informal		Light
	Target	Informal	Core	Core	Informal	Informal		Light
Data Architecture	Baseline	Core	Light	Core	Informal	Informal		Light
	Target	Informal	Core	Core	Informal	Informal		Light
Technology Architecture	Baseline	Core	Light	Core	Informal	Informal		Light
	Target	Informal	Core	Core	Informal	Informal		Light
Opportunities and Solutions		Light	Light	Light	Core	Core	Informal	Informal
Migration Planning		Light	Light	Light	Core	Core	Informal	Informal
Implementation Governance					Informal	Informal	Core	Core
Change Management		Informal	Informal	Informal	Informal	Informal	Core	Core

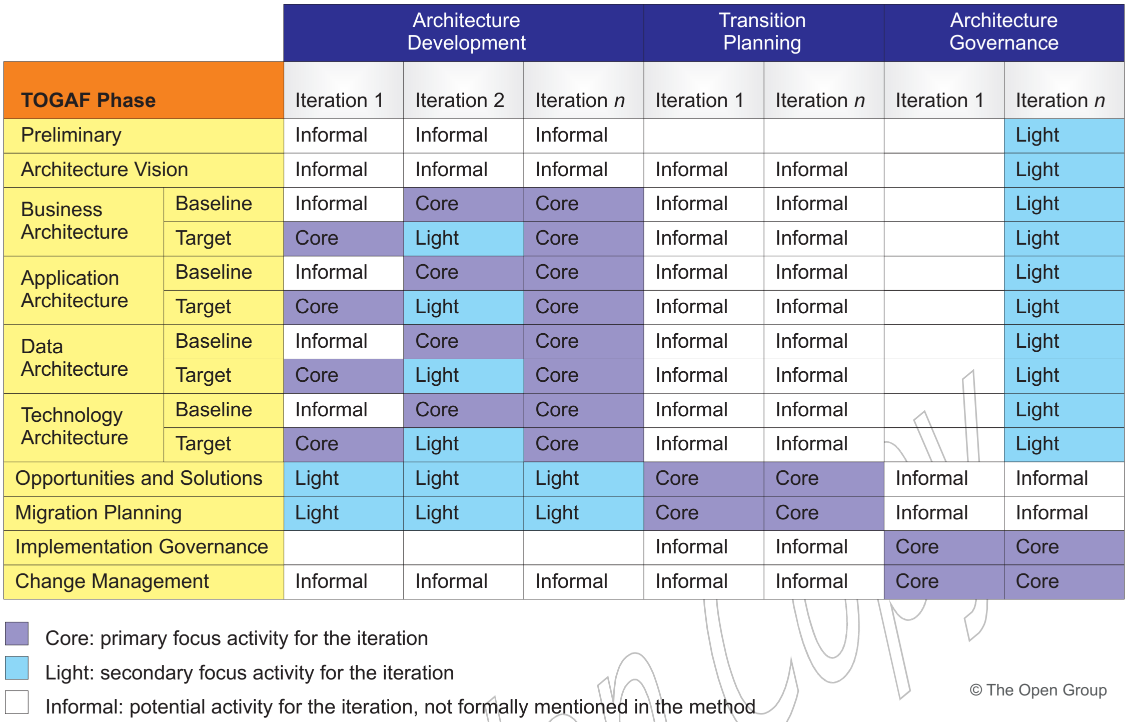
🟪 Core: primary focus activity for the iteration

🟦 Light: secondary focus activity for the iteration

□ Informal: potential activity for the iteration, not formally mentioned in the method

© The Open Group

Figure 2-4 Activity by Iteration for Baseline First Architecture Definition



		Architecture Development			Transition Planning		Architecture Governance	
TOGAF Phase		Iteration 1	Iteration 2	Iteration *n*	Iteration 1	Iteration *n*	Iteration 1	Iteration *n*
Preliminary		Informal	Informal	Informal				Light
Architecture Vision		Informal	Informal	Informal	Informal	Informal		Light
Business Architecture	Baseline	Informal	Core	Core	Informal	Informal		Light
	Target	Core	Light	Core	Informal	Informal		Light
Application Architecture	Baseline	Informal	Core	Core	Informal	Informal		Light
	Target	Core	Light	Core	Informal	Informal		Light
Data Architecture	Baseline	Informal	Core	Core	Informal	Informal		Light
	Target	Core	Light	Core	Informal	Informal		Light
Technology Architecture	Baseline	Informal	Core	Core	Informal	Informal		Light
	Target	Core	Light	Core	Informal	Informal		Light
Opportunities and Solutions		Light	Light	Light	Core	Core	Informal	Informal
Migration Planning		Light	Light	Light	Core	Core	Informal	Informal
Implementation Governance					Informal	Informal	Core	Core
Change Management		Informal	Informal	Informal	Informal	Informal	Core	Core

▨ Core: primary focus activity for the iteration

▨ Light: secondary focus activity for the iteration

□ Informal: potential activity for the iteration, not formally mentioned in the method

© The Open Group

Figure 2-5 Activity by Iteration for Target First Architecture Definition

The suggested iteration cycles mapped to the TOGAF phases are described in the following table:

Iteration Cycle	Iteration	Purpose	Description
Architecture Development (Baseline First)	Iteration 1	Define the Baseline Architecture.	This iteration comprises a pass through the Business Architecture, Information Systems Architecture, and Technology Architecture phases of the ADM, focusing on definition of the baseline. Opportunities, solutions, and migration plans are also considered to drive out the focus for change and test feasibility.

Iteration Cycle	Iteration	Purpose	Description
	Iteration 2	Define the Target Architecture and gaps.	This iteration comprises a pass through the Business Architecture, Information Systems Architecture, and Technology Architecture phases of the ADM, focusing on definition of the target and analyzing gaps against the baseline. Opportunities, solutions, and migration plans are also considered to test viability.
	Iteration *n*	Refine baseline, target, and gaps.	Subsequent Architecture Development iterations attempt to correct and refine the target to achieve an outcome that is beneficial, feasible, and viable.
Architecture Development (Target First)	Iteration 1	Define the Target Architecture.	This iteration comprises a pass through the Business Architecture, Information Systems Architecture, and Technology Architecture phases of the ADM, focusing on definition of the target. Opportunities, solutions, and migration plans are also considered to drive out the focus for change and test feasibility.
	Iteration 2	Define the Baseline Architecture and gaps.	This iteration comprises a pass through the Business Architecture, Information Systems Architecture, and Technology Architecture phases of the ADM, focusing on definition of the baseline and analyzing gaps against the target. Opportunities, solutions, and migration plans are also considered to test viability.
	Iteration *n*	Refine baseline, target, and gaps.	Subsequent Architecture Development iterations attempt to correct and refine the target to achieve an outcome that is beneficial, feasible, and viable.

Iteration Cycle	Iteration	Purpose	Description
Transition Planning	Iteration 1	Define and agree a set of improvement opportunities, aligned against a provisional Transition Architecture.	The initial iteration of Transition Planning seeks to gain buy-in to a portfolio of solution opportunities in the Opportunities & Solutions phase of ADM. This iteration also delivers a provisional Migration Plan.
	Iteration *n*	Agree the Transition Architecture, refining the identified improvement opportunities to fit.	Subsequent iterations of Transition Planning seek to refine the Migration Plan, feeding back issues into the Opportunities & Solutions phase for refinement.
Architecture Governance	Iteration 1	Mobilize Architecture Governance and change management processes.	The initial Architecture Governance iteration establishes a process for governance of change and also puts in place the appropriate people, processes, and technology to support managed access to and change of the defined architecture.
	Iteration *n*	Carry out Architecture Governance and change control.	Subsequent iterations of the Architecture Governance cycle focus on periodic reviews of change initiatives to resolve issues and ensure compliance. Results of a Change Request may trigger another phase to be revisited; for example, feeding back a new requirement to the Preliminary Phase to improve the Architecture Capability, or a new requirement for the architecture into the Architecture Development phases.

2.6 Conclusions

All of these techniques are valid applications of the ADM. Combined together, they represent how the ADM can be used in practice. The ADM should always be used in an iterative process. How this process is exercised is dependent upon organizational factors. Particular factors for consideration include:

- **The formality and nature of established process checkpoints within the organization**

  Does the organization mandate that certain groups of activities are carried out between checkpoints? Does the organization mandate that certain activities must be finalized before other activities can be carried out?

- **The level of stakeholder involvement expected within the process**

  Are stakeholders expecting to be closely involved within the development of a solution, or are they expecting to see a complete set of deliverables for review and approval?

- **The number of teams involved and the relationships between different teams**

  Is the entire architecture being developed by a specific team, or is there a hierarchy of teams with governance relationships between them?

- **The maturity of the solution area and the expected amount of rework and refinement required to arrive at an acceptable solution**

  Can the solution be achieved in a single pass, or does it require extensive proof-of-concept and prototyping work to evolve a suitable outcome?

- **Attitude to risk**

  Does the organizational culture react negatively to partially complete work products being circulated? Does the organizational culture require solutions to be proved in a trial environment before they can be implemented for mainstream application?

- **The class of engagement**

  What is the context for development of the Enterprise Architecture?

Chapter 3: Applying the ADM Across the Architecture Landscape

3.1 Overview

In a typical enterprise, many architectures will be described in the Architecture Landscape at any point in time. Some architectures will address very specific needs; others will be more general. Some will address detail; some will provide a big picture. To address this complexity, the TOGAF Standard uses the concepts of levels and the Enterprise Continuum to provide a conceptual framework for organizing the Architecture Landscape. These concepts are tightly linked with organizing actual content in the Architecture Repository and any architecture partitions discussed in the TOGAF Standard — Architecture Content.

3.2 Architecture Landscape

Levels provide a framework for dividing the Architecture Landscape into three levels of granularity:

1. **Strategic Architecture** provides an organizing framework for operational and change activity and allows for direction setting at an executive level.
2. **Segment Architecture** provides an organizing framework for operational and change activity and allows for direction setting and the development of effective architecture roadmaps at a program or portfolio level.
3. **Capability Architecture** provides an organizing framework for change activity and the development of effective architecture roadmaps realizing capability increments.

Figure 3-1 shows a summary of the classification model for Architecture Landscapes.

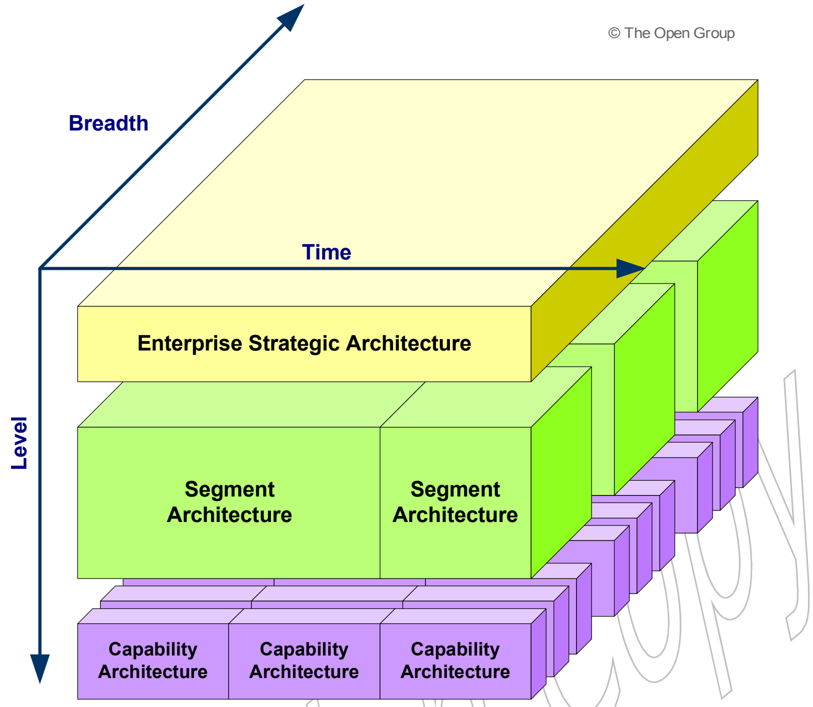


Figure 3-1 Summary Classification Model for Architecture Landscapes

The Architecture Continuum provides a method of dividing each level of the Architecture Landscape (see the TOGAF Standard — Architecture Content) by abstraction. It offers a consistent way to define and understand the generic rules, representations, and relationships in an architecture, including traceability and derivation relationships. The Architecture Continuum shows the relationships from foundation elements to organization-specific architecture, as shown in Figure 3-2.

The Architecture Continuum is a useful tool to discover commonality and eliminate unnecessary redundancy.

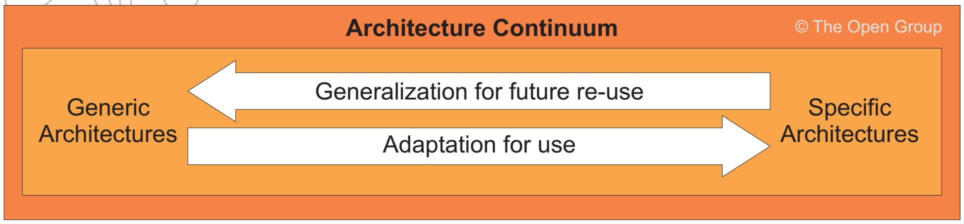


Figure 3-2 Summary of Architecture Continuum

Levels and the Architecture Continuum provide a comprehensive mechanism to describe and classify the Architecture Landscape. These concepts can be used to organize the Architecture Landscape into a set of related architectures with:

- Manageable complexity for each individual architecture or solution
- Defined groupings
- Defined hierarchies and navigation structures
- Appropriate processes, roles, and responsibilities attached to each grouping

There is no definitive organizing model for architecture, as each enterprise should adopt a model that reflects its own operating model.

3.3 Developing Architectures at Different Levels

The previous sections have identified that different types of architecture are required to address different stakeholder needs at different levels of the organization. Each architecture typically does not exist in isolation and must therefore sit within a governance hierarchy. Broad, summary architectures set the direction for narrow and detailed architectures.

A number of techniques can be employed to use the 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

At the extreme ends of the scale, either of these two options can be fully adopted. In practice, an architect is likely to need to blend elements of each to fit the exact requirements of their Request for Architecture Work. Each of these approaches is described in Chapter 2.

3.4 Organizing the Architecture Landscape to Understand the State of the Enterprise

The following characteristics are typically used to organize the Architecture Landscape:

- **Breadth**: the breadth (subject matter) area is generally the primary organizing characteristic for describing an Architecture Landscape

  Architectures are functionally decomposed into a hierarchy of specific subject areas or segments.

- **Depth**: with broader subject areas, less detail is needed to ensure that the architecture has a manageable size and complexity

  More specific subject matter areas will generally permit (and require) more detailed architectures.

- **Time**: for a specific breadth and depth an enterprise can create a Baseline Architecture and a set of Target Architectures that stretch into the future

  Broader and less detailed architectures will generally be valid for longer periods of time and can provide a vision for the enterprise that stretches further into the future.

- **Recency**: finally, each architecture view will progress through a development cycle where it increases in accuracy until finally approved

  After approval, an architecture will begin to decrease in accuracy if not actively maintained. In some cases recency may be used as an organizing factor for historic architectures.

Using the criteria above, architectures can be grouped into Strategic, Segment, and Capability Architecture levels, as described in Figure 3-1.

Chapter 4: Architecture Partitioning

4.1 Overview

Partitions are used to simplify the development and management of the Enterprise Architecture.

Partitions lie at the foundation of Architecture Governance and are distinct from levels and the organizing concepts of the Architecture Continuum (see the TOGAF Standard — Architecture Content).

Architectures are partitioned because:

- Organizational unit architectures conflict with one another
- Different teams need to work on different elements of architecture at the same time and partitions allow for specific groups of architects to own and develop specific elements of the architecture
- Effective architecture re-use requires modular architecture segments that can be taken and incorporated into broader architectures and solutions

It is impractical to present a definitive partitioning model for architecture. Each enterprise needs to adopt a partitioning model that reflects its own operating model.

This chapter discusses the classification criteria that are generally applied to architectures and how these can be leveraged to partition the enterprise into a set of architectures with manageable complexity and effective governance.

4.2 Applying Classification to Create Partitioned Architectures

For the reasons outlined in the previous section, it is valuable to partition and organize the Enterprise Continuum into a set of related solutions and architectures with:

- Manageable complexity for each individual architecture or solution
- Defined groupings
- Defined hierarchies and navigation structures
- Appropriate processes, roles, and responsibilities attached to each grouping

The following table shows how suitable classification criteria can be used to support partitioning of solutions:

Characteristic	Usage to Support Solution Partitioning
Subject Matter (Breadth)	Solutions are naturally organized into groups to support operational management and control. Examples of solution partitions according to subject matter would include applications, departments, divisions, products, services, service centers, sites, etc. Solution decomposition by subject matter is typically the fundamental technique for structuring both solutions and the architectures that represent them.
Time	Solution lifecycles are typically organized around a timeline, which allows the impact of solution development, introduction, operation, and retirement to be managed against other business activity occurring in similar time periods.
Maturity/Volatility	The maturity and volatility of a solution will typically impact the speed of execution required for the solution lifecycle. Additionally, volatility and maturity will shape investment priorities. Solutions existing in highly volatile environments may be better suited to rapid, agile development techniques.

The following table shows how each classification criteria can be used to support partitioning of architectures:

Characteristic	Usage to Support Architecture Partitioning
Depth	The level of detail within an architecture has a strong correlation to the stakeholder groups that will be interested in the architecture. Typically, less detailed architectures will be of interest to executive stakeholders. As architectures increase in detail, their relevance to implementation and operational personnel will also increase.

In practical terms, architecture discipline is used to support a number of different types of architecture that are used for different objectives. The classification criteria described above can be used in different ways to support the achievement of each objective.

The following characteristics are generally not used to partition an Architecture Landscape:

- Architectures used to describe the Architecture Landscape are generally not abstract
- Solution volatility generally prevents architectures from being defined that are far in the future; volatility also reduces the accuracy of historic architectures over time, as the organization changes and adapts to new circumstances

Using the criteria above, architectures can be grouped into partitions.

4.2.1 Activities within the Preliminary Phase

The key objective of the Preliminary Phase is to establish the Architecture Capability for the enterprise. In practical terms this activity will require the establishment of a number of architecture partitions, providing defined boundaries, governance, and ownership.

Generally speaking, each team carrying out architecture activity within the enterprise will own one or more architecture partitions and will execute the ADM to define, govern, and realize their architectures.

If more than one team is expected to work on a single architecture, this can become problematic, as the precise responsibilities of each team are difficult to establish. For this reason, it is preferable to apply partitioning to the architecture until each architecture has one owning team.

Finally, it is worth considering the distinction between standing capabilities of the enterprise and temporary teams mobilized to support a particular change initiative. Although the remit of standing teams within the enterprise can be precisely defined, it is more difficult to anticipate and specify the responsibilities of (possibly unknown) temporary architecture teams. In the cases of these temporary teams, each team should come under the governance of a standing architecture team and there should be a process within the ADM cycle of these teams to establish appropriate architecture partitioning.

Steps within the Preliminary Phase to support architecture partitioning are as follows:

- **Determine the organization structure for architecture within the enterprise**: the various standing teams that will create the architecture should be identified

  For each of these teams, appropriate boundaries should be established, including:

  - Governance bodies that are applicable to the team
  - Team membership
  - Team reporting lines

- **Determine the responsibilities for each standing architecture team**: for each architecture team, the responsibilities should be identified

  This step applies partitioning logic to the Enterprise Architecture in order to firstly identify the scope of each team and secondly to partition the architecture under the remit of a single team. Once complete, this step should have partitioned the entire scope of the enterprise and should have assigned responsibility for each partitioned architecture to a single team. Partitioning should create a definition of each architecture that includes:

  - Subject matter areas being covered
  - Level of detail at which the team will work
  - Time periods to be covered
  - Stakeholders

- **Determine the relationships between architectures**: once a set of partitioned architectures has been created, the relationships between architectures should be developed

  This step allows governance relationships to be formalized and also shows where artifacts from one architecture are expected to be re-used within other architectures.

Areas of consideration include:

- Where do different architectures overlap/dovetail/drill-down?
- What are the compliance requirements between architectures?

Once the Preliminary Phase is complete, the teams conducting the architecture should be understood. Each team should have a defined scope and the relationships between teams and architecture should be understood. Allocation of teams to architecture scope is illustrated in Figure 4-1.

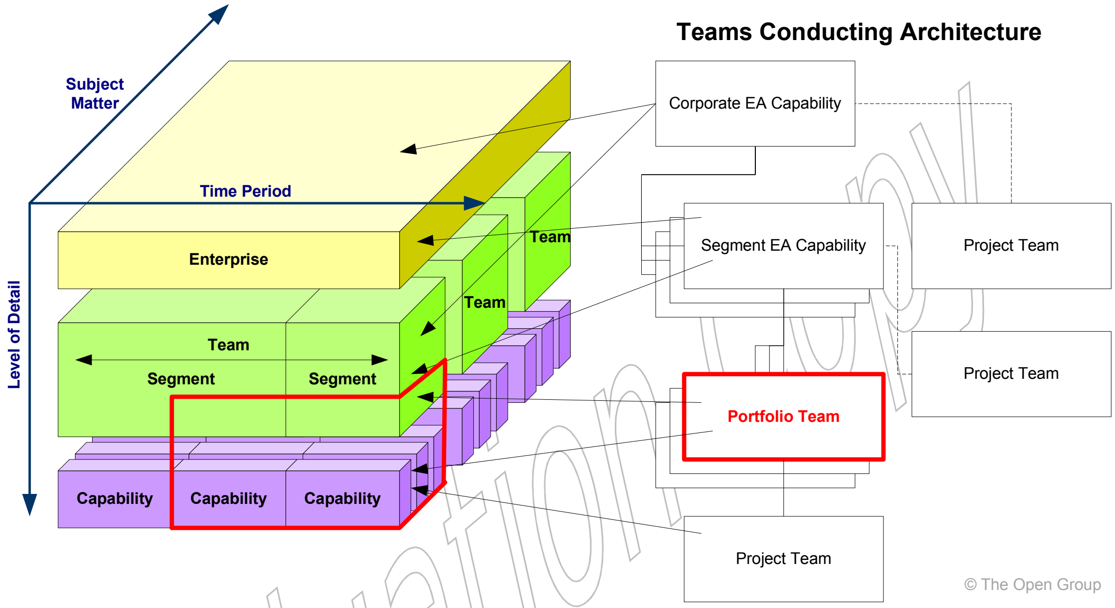


Figure 4-1 Allocation of Teams to Architecture Scope

4.3 Integration

The creation of partitioned architectures runs the risk of producing a fragmented and disjointed collection of architectures that cannot be integrated to form an overall big picture (see the TOGAF Standard — Architecture Development Method).

For large complex enterprises, federated architectures — independently developed, maintained, and managed architectures that are subsequently integrated within an integration framework — are typical. Federated architectures typically are used in governments and conglomerates, where the separate organizational units need separate architectures. Such a framework specifies the principles for interoperability, migration, and conformance. This allows specific business units to have architectures developed and governed as stand-alone architecture projects. More details and guidance on specifying the interoperability requirements for different solutions can be found in the TOGAF Standard — ADM Techniques.

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. Content frameworks, such as the TOGAF content framework (refer to the TOGAF Standard — Architecture Content) can be used to specify standard building blocks and artifacts that are the subject of content integration standards.

For example, a standard catalog of business processes can be agreed for an enterprise. Subsequent architectures can then ease integration by using the same process list and cross-referencing other aspects of the architecture to those standard processes.

Integration can be addressed from a number of dimensions:

- Integration across the architecture 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

Figure 4-2 shows how architectural content can be aggregated using a variety of techniques.

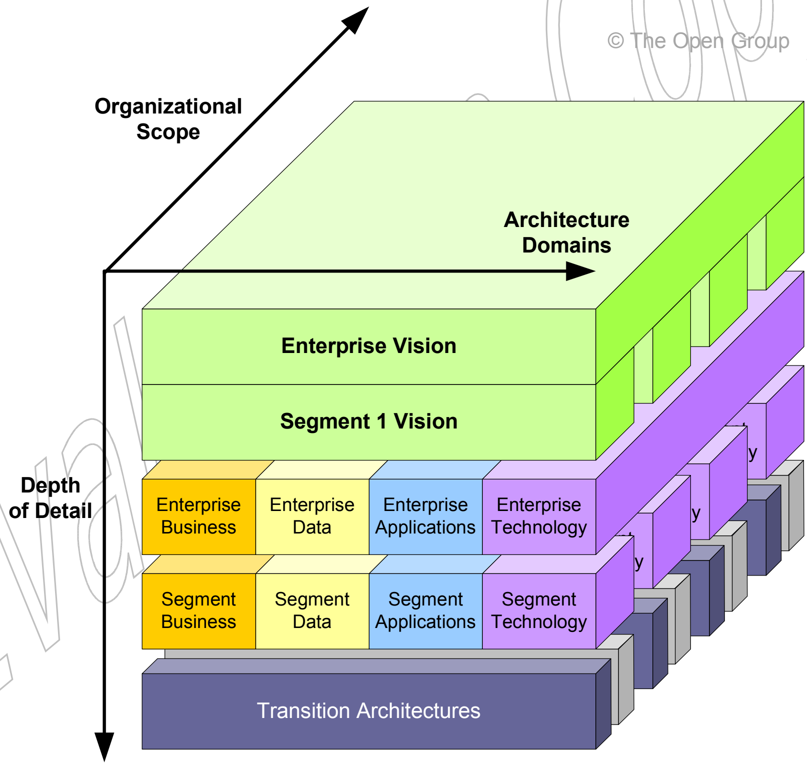


Figure 4-2 Architecture Content Aggregation

Index