

The Open Group Standard

TOGAF[®] Standard — Introduction and Core Concepts

The Open Group

Copyright © 2005-2022, The Open Group

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

Any use of this publication for commercial purposes is subject to the terms of the Annual Commercial License relating to it. For further information, see www.opengroup.org/legal/licensing.

The Open Group Standard

TOGAF® Standard — Introduction and Core Concepts

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

Chapter 1	Introduction	1
1.1	Executive Overview	2
1.2	Structure of this Document	6
1.3	Information on Using the TOGAF Standard	6
1.3.1	Conditions of Use	6
1.3.2	How Much Does the TOGAF Standard Cost?	7
1.3.3	Downloads	7
1.4	Why Join The Open Group?	7
Chapter 2	The TOGAF Documentation Set	9
2.1	Structure of the TOGAF Documentation Set	9
2.2	The TOGAF Standard	10
2.3	The TOGAF Library	13
Chapter 3	Core Concepts.....	15
3.1	What is the TOGAF Standard?	15
3.2	What is Architecture in the Context of the TOGAF Standard?	15
3.3	What Kind of Architecture Does the TOGAF Standard Deal With?	15
3.4	Architecture Development Method.....	16
3.5	Enterprise Architecture Services	18
3.5.1	Enterprise Support Services.....	20
3.5.2	Design Support Services	20
3.5.3	Development Support Services	20
3.5.4	Requirements Elicitation and Understanding Services	20
3.5.5	Architecture Planning Services.....	21
3.5.6	Enterprise Architecture Practice Development Support Services	21
3.6	Deliverables, Artifacts, and Building Blocks	21
3.7	Architecture Abstraction.....	23
3.7.1	Contextual Abstraction Level	24
3.7.2	Conceptual Abstraction Level	24
3.7.3	Logical Abstraction Level	24
3.7.4	Physical Abstraction Level	24
3.8	Architecture Principles	24
3.9	Interoperability	25

3.10	Enterprise Continuum	26
3.11	Architecture Repository	27
3.12	TOGAF Content Framework and Enterprise Metamodel	29
3.12.1	Overview	29
3.12.2	Content Framework	29
3.12.3	Enterprise Metamodel	31
3.12.4	Developing the Enterprise Metamodel	32
3.13	Establishing and Maintaining an Enterprise Architecture Capability	33
3.14	Establishing the Architecture Capability as an Operational Entity	34
3.15	Using the TOGAF Standard with Other Frameworks	35
3.16	Using the TOGAF Framework with Different Architecture Styles	36
3.17	Architecture Views and Viewpoints	37
3.18	Enterprise Agility	38
3.19	Risk Management	39
Chapter 4	Definitions	41
Appendix A	Referenced Documents	55
Appendix B	Glossary of Supplementary Definitions	61
Appendix C	Abbreviations	69
	Index	73

List of Figures

2-1	The TOGAF Documentation Set	9
2-2	Structure of the TOGAF Standard	10
2-3	The TOGAF Standard	12
2-4	The TOGAF Library Continuum	14
3-1	Architecture Development Cycle	16
3-2	Relationships between Deliverables, Artifacts, and Building Blocks	22
3-3	Example — Architecture Definition Document	22
3-4	Enterprise Continuum	27
3-5	TOGAF Architecture Repository Structure	28
3-6	Content Framework by ADM Phase	30
3-7	Applying the Enterprise Continuum	32
3-8	The TOGAF Core Enterprise Metamodel	33
3-9	TOGAF Architecture Capability Overview	33
3-10	Basic Architectural Concepts	37

Contents

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.

This Document

This is the TOGAF Standard — Introduction and Core Concepts.

This document is part of the TOGAF Standard, and provides an introduction to the standard, including an executive overview of Enterprise Architecture, a description of how the standard is organized, and a summary of core concepts. It also contains the material common to the individual documents that comprise the standard, such as the definitions, as well as document references and abbreviations.

The TOGAF Documentation

The TOGAF documentation set comprises a portfolio of documents, built around the TOGAF Standard.

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 Standard is presented as a series of free-standing, but closely linked documents and is supplemented by an extensive and growing portfolio of guidance material, providing practical guidance in the application of the TOGAF Standard in specific contexts.

The TOGAF Standard is a standard of The Open Group. The Open Group works with customers and suppliers of technology products and services, and with consortia and other standards organizations to capture, clarify, and integrate current and emerging requirements, establish standards and policies, and share best practices. Standards ensure openness, interoperability, and consensus.

At the time of publication, the TOGAF Standard comprises the following documents:

- TOGAF Standard — Introduction and Core Concepts
- TOGAF Standard — Architecture Development Method
- TOGAF Standard — ADM Techniques
- TOGAF Standard — Applying the ADM
- TOGAF Standard — Architecture Content
- TOGAF Standard — Enterprise Architecture Capability and Governance
- TOGAF Standard — TOGAF Series Guides (set of documents)

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.

Other audiences are Digital and Agile Practitioners, Product Managers, and C-Suite. These audiences will find more detailed guidance on how to apply the standard to fulfill specific needs in the TOGAF Series Guides set of documents.

1. The TOGAF Library is a publicly-accessible resource located at www.opengroup.org/togaf-library.

Keywords

architecture, architecture framework, architecture development method, architect, architecting, enterprise architecture, enterprise architecture framework, enterprise architecture method, method, methods, open, group, technical reference model, standards library

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.

CMMI is a registered trademark of CMMI Institute.

COBIT is a registered trademark of the Information Systems Audit and Control Association (ISACA) and the IT Governance Institute.

FICO is a registered trademark of Fair Isaac Corporation in the United States and other countries.

IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers, Inc.

ITIL, MSP, and PRINCE2 are registered trademarks of AXELOS Limited.

Java is a registered trademark of Oracle and/or its affiliates.

MDA, Model-Driven Architecture, Object Management Group, OMG, SysML, and UML are registered trademarks and BMM, BPMN, Business Motivation Model, Business Process Modeling Notation, ITPMF, IT Portfolio Management Facility, SPEM, Systems Modeling Language, and Unified Modeling Language are trademarks of the Object Management Group.

Merriam-Webster Collegiate Dictionary is a registered trademark of Merriam-Webster, Incorporated.

OASIS is a registered trademark of OASIS, the open standards consortium.

PMBOK is a registered trademark of the Project Management Institute, Inc. which is registered in the United States and other nations.

Zachman is a registered trademark of Zachman International, Inc.

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.

Participants

The TOGAF Standard, 10th Edition was prepared by The Open Group Architecture Forum. When The Open Group approved this standard for publication in April 2022, the Architecture Forum Officers were as follows:

Mick Adams, EY, Co-Chair

Paul Homan, IBM, Co-Chair (2018-2021)

Céline Lescop, AXA, Co-Chair (from March 2021)

Sonia Gonzalez, The Open Group, TOGAF Product Manager

Mark Dickson, The Open Group, Architecture Forum Director

Daniel Hutley, The Open Group, Architecture Forum Director

Architecture Forum Technical Reviewers

Technical reviewers are those individuals who have submitted comments during the Company Review, or participated in an issue resolution meeting during the development of the TOGAF Standard, 10th Edition (arranged in alphabetical order).

(Please note affiliations were current at the time of approval.)

- Adam Art, Raytheon Technologies
- Graham Berrisford, Avancier
- Terence Blevins, Enterprise Wise LLC & Fellow of The Open Group
- Ken Block, Raytheon Technologies
- Steve Dupont, Boeing
- Ivan Eikenberry, Raytheon Technologies
- Chris Forde, Vice-President Enterprise Architecture, The Open Group
- Mats Gejnevall, Biner Consulting
- David Gilmour, Mundo Cognito
- Susan Gottschlich, Raytheon Technologies
- Heidi Hasz, Global Knowledge Training LLC
- Judith Jones, ATE Enterprises
- Andrew Josey, VP Standards & Certification, The Open Group
- Rolf Knoll, Novatec Consulting GmbH
- Bryan Lail, Raytheon Technologies

Participants

- Mike Lambert, Fellow of The Open Group
- Frédéric Lé, DXC
- Seshu Madabhushi, Tata Consultancy
- Stephen Marshal, IBM
- Terence McKiernan, Raytheon Technologies
- Jon McLeod, EA Learning Pty Ltd.
- Marc Perna, Raytheon Technologies
- David Pham, Raytheon Technologies
- Luke Ranck, Raytheon Technologies
- James Renfro, Raytheon Technologies
- G. Edward Roberts, Elparazim
- Naveen Sharma, HCL Technologies
- Ralph Shaw, Raytheon Technologies
- Dirk Vollmerhaus, Scape Consulting GmbH
- Robert Weisman, Build the Vision
- Mark Whitbread, Raytheon Technologies

Architecture Forum Members

An up-to-date list of Forum members can be found at: www.opengroup.org/architecture.

Acknowledgements

The Open Group gratefully acknowledges The Open Group Architecture Forum for developing the TOGAF Standard.

The Open Group gratefully acknowledges the significant contribution to the evolution of this release of the TOGAF Standard by Mike Lambert, Fellow of The Open Group and Paul Homan, IBM.

The Open Group gratefully acknowledges those past and present members of the Architecture Forum who have served as its Officers (Chairs and Vice-Chairs) since its inception. In alphabetical order:

Mick Adams	Céline Lescop
Christer Askerfjord	Stuart Macgregor
Terence Blevins	Ian McCall
Bill Estrem	Tara Paider
Hugh Fisher	Barry Smith
Chris Forde	Walter Stahlecker
Chris Greenslade	Sheena Thompson
Ed Harrington	Paul van der Merwe
Peter Haviland	Dave Van Gelder
Paul Homan	Jane Varnus
Dave Hornford	Vish Viswanathan
David Jackson	Robert Weisman
Mike Lambert	Hal Wilson

Chapter 1: Introduction

The TOGAF Standard is a framework for Enterprise Architecture. It may be used freely by any organization wishing to develop an Enterprise Architecture for use within that organization (see [Section 1.3.1](#)).

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

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

This version builds on previous versions of the TOGAF Standard and updates the material available to architecture practitioners to assist them in building a sustainable Enterprise Architecture. Work on White Papers and Guides describing how to integrate and use this standard with other frameworks and architectural styles has highlighted the universal framework parts of the standard, as well as industry, architecture style, and purpose-specific tools, techniques, and guidance. This work is embodied in the TOGAF Library.¹

Although all of the TOGAF documentation works together as a whole, it is expected that organizations will customize it during adoption, and deliberately choose some elements, customize some, exclude some, and create others. For example, an organization may wish to adopt the TOGAF metamodel, but elect not to use any of the guidance on how to develop an in-house Technology Architecture because they are heavy consumers of cloud services.

You are recommended to first read the Executive Overview (see [Section 1.1](#)), which includes an outline of The Open Group understanding of Enterprise Architecture and answers to fundamental questions, such as:

- Why is an Enterprise Architecture needed?
- Why use the TOGAF Standard as a framework for Enterprise Architecture?

1. The TOGAF Library provides an online publicly available structured list of Guides, White Papers, and other resources. Refer to TOGAF Library at www.opengroup.org/togaf-library.

1.1 Executive Overview

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

What is an enterprise?

The TOGAF Standard considers an "enterprise" to be any collection of organizations that have common goals.

For example, an enterprise could be:

- A whole corporation or a division of a corporation
- A government agency or a single government department
- A chain of geographically distant organizations linked together by common ownership
- Groups of countries, governments, or governmental organizations (such as militaries) working together to create common or shareable deliverables or infrastructures
- Partnerships and alliances of businesses working together, such as a consortium or supply chain

The term "Enterprise" in the context of "Enterprise Architecture" can be applied to either an entire enterprise, encompassing all of its business activities and capabilities, information, and technology that make up the entire infrastructure and governance of the enterprise, or to one or more specific areas of interest within the enterprise. An enterprise may include partners, suppliers, and customers as well as internal business units. In all cases, the architecture crosses multiple systems, and multiple functional groups within the enterprise.

The enterprise operating model concept is useful to determine the nature and scope of the Enterprise Architecture within an organization. Many organizations may comprise multiple enterprises, and may develop and maintain a number of independent Enterprise Architectures to address each one. These enterprises often have much in common with each other including processes, functions, and their information systems, and there is often great potential for wider gain in the use of a common architecture framework. For example, a common framework can provide a basis for the development of common building blocks and solutions, and a shareable Architecture Repository for the integration and re-use of business models, designs, information, and data.

Why is an Enterprise Architecture needed?

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

The effective management and exploitation of information and Digital Transformation are key factors to business success, and indispensable means to achieving competitive advantage. An Enterprise Architecture addresses this need, by providing a strategic context for the evolution and reach of digital capability in response to the constantly changing needs of the business environment.

Furthermore, a good Enterprise Architecture enables you to achieve the right balance between business transformation and continuous operational efficiency. It allows individual business units

to innovate safely in their pursuit of evolving business goals and competitive advantage. At the same time, the Enterprise Architecture enables the needs of the organization to be met with an integrated strategy which permits the closest possible synergies across the enterprise and beyond.

And lastly, much of the global privacy legislation demands that processes around personal data are fully documented in a way that can be easily understood by untrained readers — such as the data subjects and judges and lawyers. The penalties for failing to have this can be very significant. Clearly the creation of this basic documentation arises from the changed fundamental considerations and this is now crucial.

What are the benefits of an Enterprise Architecture?

An effective Enterprise Architecture can bring important benefits to the organization. Potential benefits of an Enterprise Architecture include:

- More effective strategic decision-making by C-Level executives and business leaders:
 - Quick response to change and support for enterprise agility aligned with the organization strategy
 - Organizational transformation, adopting new trends in business and technology
 - Organizational change to support Digital Transformation
 - Organizational and operating model changes to improve efficiency and effectiveness
- More effective and efficient business operations:
 - Lower business operation costs
 - More agile organization
 - Business capabilities shared across the organization
 - Lower change management costs
 - More flexible workforce
 - Improved business productivity
 - Improved organization integration in support of mergers and acquisitions
- More effective and efficient Digital Transformation and operations:
 - Extending effective reach of the enterprise (e.g., through digital capability)
 - Bringing all components of the enterprise into a harmonized environment
 - Lower development, deployment, operations, support, and maintenance costs
 - Improved interoperability
 - Improved system management
 - Improved ability to address critical enterprise-wide issues (e.g., security)
 - Easier upgrade and exchange of system components

- Better return on existing investment, reduced risk for future investment:
 - Reduced complexity in the business and IT
 - Maximized return on investment in existing business and IT
 - The flexibility to make, buy, or outsource business and IT solutions
 - Understanding how return on investment changes over time
- Faster, simpler, and cheaper procurement:
 - Buying decisions are simpler, because the information governing procurement is readily available in a coherent plan
 - The procurement process is faster — maximizing procurement speed and flexibility without sacrificing architectural coherence
 - The ability to procure heterogeneous, multi-vendor open systems
 - The ability to secure more economic capabilities

What specifically would prompt the development of an Enterprise Architecture?

The reasons for embarking on an Enterprise Architecture review or development are varied, including:

- Business-driven initiatives to enable business transformation; for example, to leverage digital services and products as revenue generating assets
- Technology-driven initiatives for efficiency and cost reduction; for example, technology consolidation initiatives, where the consolidation destination can be physical, virtual, or a combination
- Merger or acquisition, where the return on investment is only realized after technology efficiencies are realized
- Management of technical debt accrued by agile development initiatives

In all of these situations, Enterprise Architecture review or development is needed to manage complexity when change involves multiple systems with multiple inter-dependencies.

Often key people identify areas of change required in order for new business goals to be met. Such people are commonly referred to as the "stakeholders" in the change. The role of the architect is to address their concerns by:

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

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

What is an architecture framework?

An architecture framework is a foundational structure, or set of structures, which can be used for developing a broad range of different architectures. It should include a method for describing both a baseline and target state of the enterprise, in terms of a set of building blocks for showing how the building blocks fit together and planning the evolution from baseline to target states.

A framework is typically tailored to meet the specific needs of the organization. Tailoring of the framework should establish a set of tools and a common vocabulary.

Why use the TOGAF Standard as a framework for Enterprise Architecture?

The TOGAF Standard has been developed through the collaborative efforts of the whole community. Using the TOGAF Standard results in Enterprise Architecture that is consistent, reflects the needs of stakeholders, employs best practice, and gives due consideration both to current requirements and the perceived future needs of the business.

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

The TOGAF Standard value proposition is to enable organizations to operate in an efficient and effective way using a proven and recognized set of best practices, across the enterprise and in different sectors to address specific business and technology trends.

A key consideration is that guidance provided by the standard is intended to be adapted to address different needs and particular use-cases. That means it can be used to create a sustainable Enterprise Architecture for a broad range of use-cases, including agile enterprises and Digital Transformation.

Who would benefit from using the TOGAF Standard?

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

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

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

Organizations that need to adapt to face new business and market challenges, to improve value propositions to their customers as part of Digital Transformation.

When should Enterprise Architecture be done?

To get the greatest benefit from Enterprise Architecture it should be done early and throughout the change process to help decision-makers understand the implications of their decisions. Without this understanding, costly mistakes can be made and Enterprise Architecture is not serving its fullest potential. Enterprise Architecture done after decisions are made is merely documentation of those decisions or at best enforcement of those decisions. No insight is gained as to the effect of those decisions which could be far-reaching and perhaps detrimental.

1.2 Structure of this Document

This document introduces the TOGAF Standard and the TOGAF Library, and includes definitions and referenced materials relevant to the individual elements of the Standard.

- This chapter provides a general introduction to Enterprise Architecture and the TOGAF Standard
- Chapter 2 describes:
 - The scope and structure of the materials that make up the TOGAF Standard
 - The scope and structure of the TOGAF Library
- Chapter 3 describes the core concepts that are used across the components of the TOGAF Standard
- Chapter 4 contains definitions of terms that are used consistently across the components of the TOGAF Standard
- Appendix A contains a list of documents referenced in the TOGAF Standard
- Appendix B contains a supplementary list of definitions of terms that may be encountered when reading the materials that make up the TOGAF Standard
- Appendix C contains a list of commonly used abbreviations

1.3 Information on Using the TOGAF Standard**1.3.1 Conditions of Use**

The TOGAF Standard is freely available for viewing online without a license. Alternatively, it can be downloaded and stored under license, as explained on the TOGAF information website.

In either case, the TOGAF Standard can be used freely by any organization wishing to do so to develop an architecture for use within that organization. No part of it may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, for any other purpose including, but not by way of limitation, any use for commercial gain, without the prior permission of the copyright owners.

1.3.2 How Much Does the TOGAF Standard Cost?

The Open Group is committed to delivering greater business efficiency by bringing together buyers and suppliers of information systems to lower the barriers of integrating new technology across the enterprise. Its goal is to realize the vision of Boundaryless Information Flow.

The TOGAF Standard is a key part of its strategy for achieving this goal, and The Open Group wants it to be taken up and used in practical architecture projects, and the experience from its use fed back to help improve it.

The Open Group therefore publishes it on its public web server, and allows and encourages its reproduction and use free-of-charge by any organization wishing to use it internally to develop an Enterprise Architecture. (There are restrictions on its commercial use, however; see [Section 1.3.1.](#))

1.3.3 Downloads

Downloads of the TOGAF Standard, including printable PDF files, are available under license from the TOGAF information website (refer to www.opengroup.org/togaf/downloads). The license is free to any organization wishing to use the standard entirely for internal purposes (for example, to develop an Enterprise Architecture for use within that organization).

1.4 Why Join The Open Group?

Organizations wishing to reduce the time, cost, and risk of implementing multi-vendor solutions that integrate within and between enterprises need The Open Group as their key partner.

The Open Group brings together the buyers and suppliers of information systems worldwide, and enables them to work together, both to ensure that IT solutions meet the needs of customers, and to make it easier to integrate IT across the enterprise. The TOGAF Standard is a key enabler in this task.

Yes, the TOGAF Standard itself is freely available. But how much will you spend on developing or updating your Enterprise Architecture? And how much will you spend on procurements based on that architecture? The price of membership of The Open Group is insignificant in comparison with these amounts.

In addition to the general benefits of membership, as a member of The Open Group you will be eligible to participate in The Open Group Architecture Forum, which is the development program within which the TOGAF Standard is evolved, and in which TOGAF users come together to exchange information and feedback. Members of the Architecture Forum gain:

- Immediate access to the fruits of the current TOGAF work program (not publicly available until publication of the next edition of the TOGAF Standard) — in effect, the latest information on the standard
- Exchange of experience with other customer and vendor organizations involved in Enterprise Architecture in general, and networking with architects using the TOGAF Standard in significant architecture development projects around the world
- Peer review of specific architecture case study material

Chapter 2: The TOGAF Documentation Set

2.1 Structure of the TOGAF Documentation Set

The TOGAF documentation set consists of a portfolio of documents illustrated in [Figure 2-1](#).

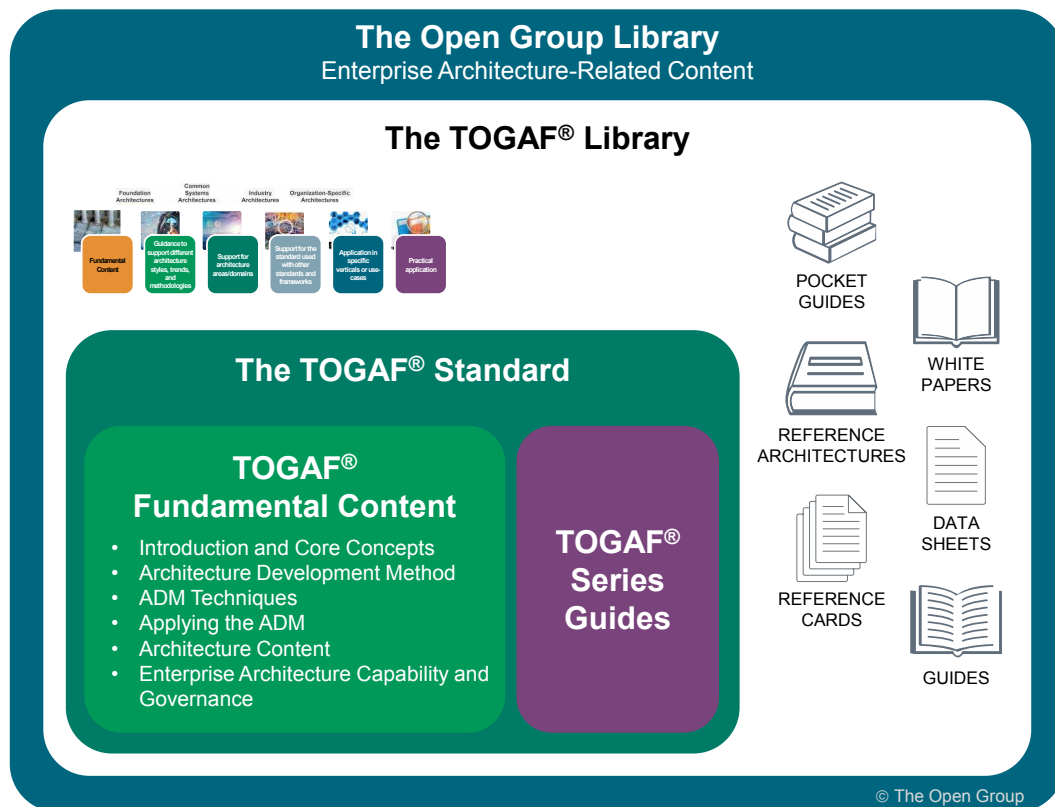


Figure 2-1 The TOGAF Documentation Set

The TOGAF Standard

The TOGAF Standard describes the generally applicable approach to Enterprise and IT Architecture. It is presented as a series of free-standing, but closely linked documents, as shown in [Figure 2-1](#).

The TOGAF Standard is a standard of The Open Group. The Open Group works with customers and suppliers of technology products and services, and with consortia and other standards organizations to capture, clarify, and integrate current and emerging requirements, establish standards and policies, and share best practices. Standards ensure openness, interoperability, and consensus.

The TOGAF Library

The TOGAF Library is a portfolio of additional guidance material, which supports the practical application of the TOGAF approach.

2.2 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, an extensive and growing portfolio of guidance material, providing practical guidance in the application of the TOGAF framework in specific contexts.

The structure of the TOGAF Standard reflects the structure and content of an Architecture Capability within an enterprise, as shown in [Figure 2-2](#).

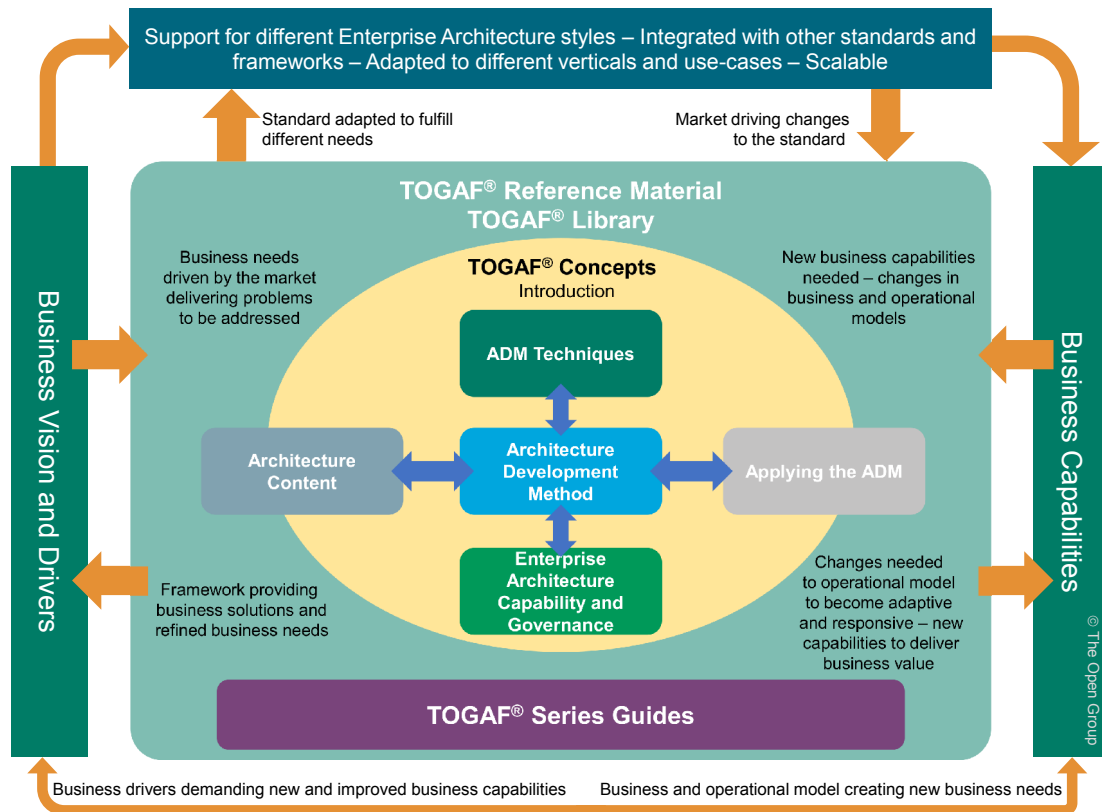


Figure 2-2 Structure of the TOGAF Standard

The TOGAF Standard Fundamental Content is presented as six free-standing documents:

- The TOGAF Standard — Introduction and Core Concepts (this document)
- The TOGAF Standard — Architecture Development Method

This document describes the TOGAF Architecture Development Method (ADM) — an iterative approach to developing an Enterprise Architecture.
- The TOGAF Standard — ADM Techniques

This document contains a collection of techniques available for use in applying the TOGAF approach and the TOGAF ADM.
- The TOGAF Standard — Applying the ADM

This document contains guidelines for adapting the TOGAF ADM to address the specific style of architecture required in a practical context.
- The TOGAF Standard — Architecture Content

This document describes the TOGAF Content Framework and a structured metamodel for architectural artifacts, the use of re-usable Architecture Building Blocks (ABBs), and an overview of typical architecture deliverables.

■ The TOGAF Standard — Enterprise Architecture Capability and Governance

This document discusses the organization, processes, skills, roles, and responsibilities required to establish and operate an architecture function within an enterprise and describes an Enterprise Architecture governance framework.

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

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

The TOGAF Standard comprises the TOGAF Fundamental Content and a collection of TOGAF Series Guides, which provide the practical guidance in the application of the TOGAF Standard.

Figure 2-3 depicts the structure and scope of the TOGAF Standard.

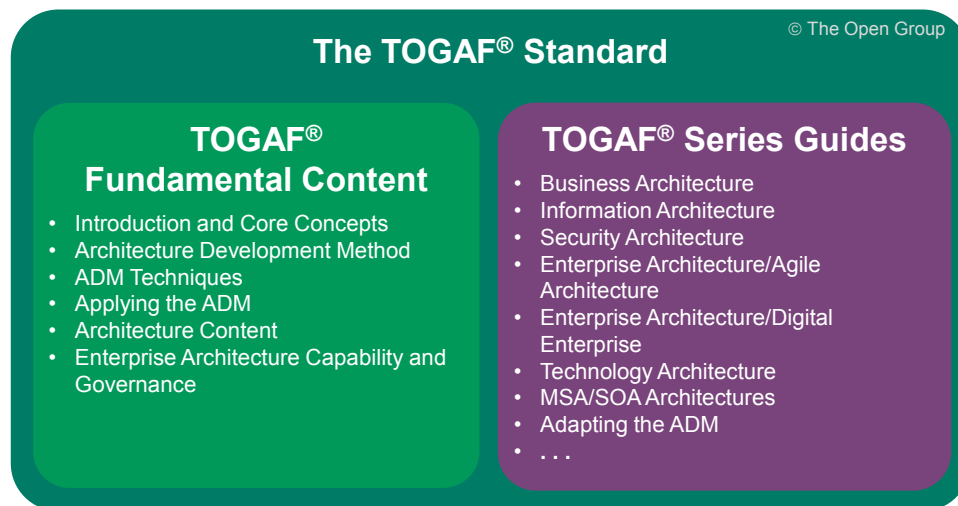


Figure 2-3 The TOGAF Standard

Besides the core framework content covered by the six documents explained above, the standard provides guidance to address specific concerns and use-cases through the TOGAF Series Guides.

The TOGAF Series Guides are designed to support more specific needs from practitioners who need further explanation or more detail than that provided in the core content.

Not all the TOGAF Series Guides will be relevant in every situation. However, Enterprise

Architects who are planning the deployment of the TOGAF Standard should be aware of the guidance available.

This content will evolve more rapidly than the core content to cover new needs as they emerge from market trends and the needs of the industry. There is a set of activities running continuously in The Open Group Architecture Forum to deliver this content following a continuous and incremental delivery pipeline.

Examples of the areas covered by this guidance material are:

- Strategy decision-making and business value-oriented decisions
- Business Architecture and operating model description
- Information and data management
- Information system guidance
- Information reference models and data integration models
- Technology Architecture: how Enterprise Architecture as a practice can be applied to adopt new technology trends to assess if the organization owns the right capabilities to support the new technology adoption
- Security Architecture: how the TOGAF Standard can be applied to deliver and support Security Architecture and risk management
- How Enterprise Architecture as a practice and the TOGAF Standard can be applied to support the agile enterprise, to be delivered following an agile style, and how the standard can support organizations using agile methodologies
- How Enterprise Architecture as a practice and the TOGAF Standard support the digital enterprise so that organizations can deliver digital products, and improve their digital offering and digital value proposition
- How the standard can be applied with other standards and methodologies of The Open Group such as the O-AA™ Standard, the DPBoK™ Standard, the IT4IT™ Reference Architecture, the ArchiMate® Specification, Microservices Architecture (MSA), security standards, and also with other standards bodies' standards and best practices

The complete set of TOGAF Series Guides can be found at www.opengroup.org/library/guides/togaf/togaf-series-guides.

2.3 The TOGAF Library

Accompanying the TOGAF Standard is a broad portfolio of guidance material, known as the TOGAF Library,² to support the practical application of the TOGAF approach. The TOGAF Library is a reference library containing guidelines, templates, patterns, and other forms of reference material to accelerate the creation of new architectures for the enterprise.

The TOGAF Library is maintained under the governance of The Open Group Architecture Forum.

The TOGAF Library supports one of the classes of information in the TOGAF Enterprise Architecture Repository, the Reference Library.

2. The TOGAF Library is a publicly accessible resource located at www.opengroup.org/togaf-library.

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 TOGAF Library is a Reference Library of potentially useful resources. The TOGAF Library follows a categorization model based on capabilities and features that can be delivered into the market through different sets of documents and resources, as depicted in [Figure 2-4](#).



Figure 2-4 The TOGAF Library Continuum

Chapter 3: Core Concepts

For the purposes of the TOGAF Standard, the core concepts provided in this chapter apply.

3.1 What is the TOGAF Standard?

The TOGAF Standard is an architecture framework. It provides the methods and tools for assisting in the acceptance, production, use, and maintenance of an Enterprise Architecture. It is based on an iterative process model supported by best practices and a re-usable set of existing architecture assets. See [Section 2.2](#).

3.2 What is Architecture in the Context of the TOGAF Standard?

ISO/IEC/IEEE 42010:2011 defines "architecture" as:

"The fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution."

The TOGAF Standard embraces but does not strictly adhere to ISO/IEC/IEEE 42010:2011 terminology. In addition to the ISO/IEC/IEEE 42010:2011 definition of "architecture", the TOGAF Standard defines a second meaning depending upon the context:

"The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time."

The TOGAF Standard considers the enterprise as a system and endeavors to strike a balance between promoting the concepts and terminology drawn from relevant standards, and commonly accepted terminology that is familiar to the majority of the TOGAF readership. For more on terminology, refer to [Chapter 4](#) and the TOGAF Standard — Architecture Content.

3.3 What Kind of Architecture Does the TOGAF Standard Deal With?

There are four architecture domains that are commonly accepted as subsets of an overall Enterprise Architecture, all of which the TOGAF Standard is designed to support:

- The **Business Architecture** defines the business strategy, governance, organization, and key business processes
- The **Data Architecture** describes the structure of an organization's logical and physical data assets and data management resources

- The **Application Architecture** provides a blueprint for the individual applications to be deployed, their interactions, and their relationships to the core business processes of the organization
- The **Technology Architecture** describes the digital architecture and the logical software and hardware infrastructure capabilities and standards that are required to support the deployment of business, data, and applications services. This includes digital services, Internet of Things (IoT), social media infrastructure, cloud services, IT infrastructure, middleware, networks, communications, processing, standards, etc.

There are many other domains that could be defined by combining appropriate views of the Business, Data, Application, and Technology domains. For example:

- Information Architecture
- Risk and Security Architectures
- Digital Architecture

The TOGAF framework enables the creation of these multi-dimensional views and categorizes them to create specific domains that enable an enterprise to consider the wider scope of their enterprise and capabilities.

3.4 Architecture Development Method

The TOGAF Architecture Development Method (ADM) provides a tested and repeatable process for developing architectures. The ADM includes establishing an architecture framework, developing architecture content, transitioning, and governing the realization of architectures.

All of these activities are carried out within an iterative cycle of continuous architecture definition and realization that allows organizations to transform their enterprises in a controlled manner in response to business goals and opportunities. This is illustrated in [Figure 3-1](#).

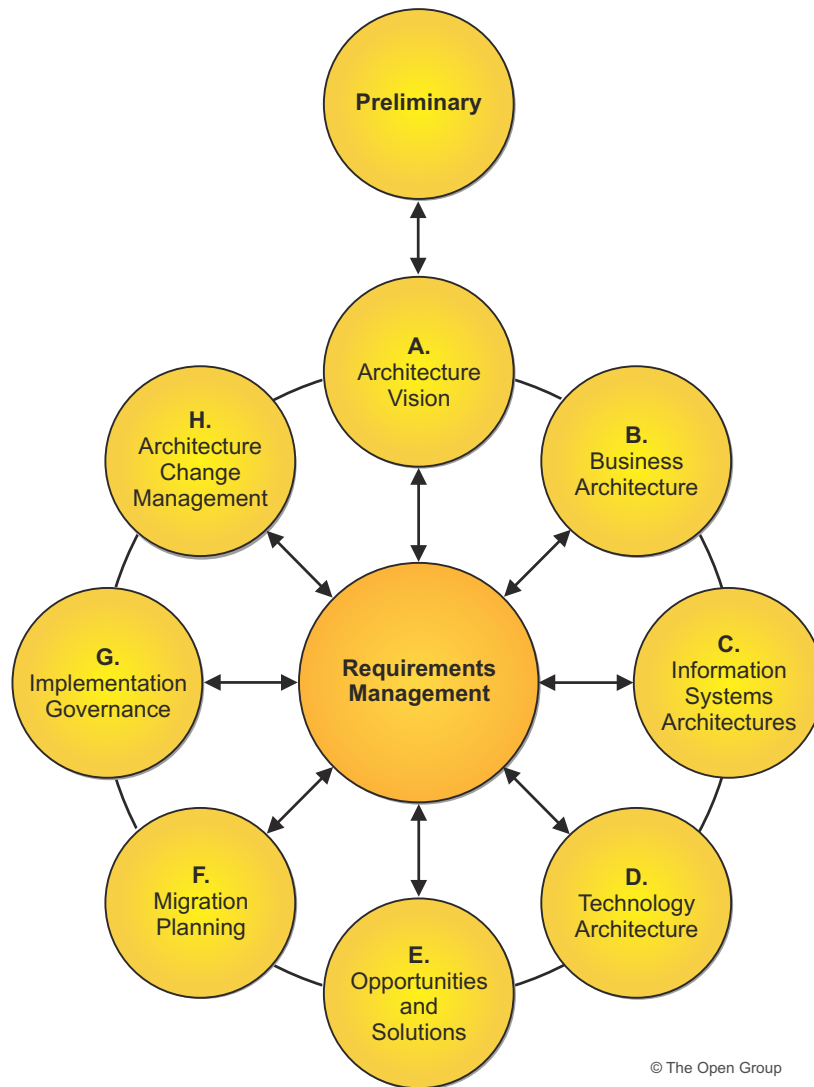


Figure 3-1 Architecture Development Cycle

Phases within the ADM are as follows:

- The **Preliminary Phase** describes the preparation and initiation activities required to create an Architecture Capability including customization of the TOGAF framework and definition of Architecture Principles
- **Phase A: Architecture Vision** describes the initial phase of an architecture development cycle

It includes information about defining the scope of the architecture development initiative, identifying the stakeholders, creating the Architecture Vision, and obtaining approval to proceed with the architecture development.

- **Phase B: Business Architecture** describes the development of a Business Architecture to support the agreed Architecture Vision
- **Phase C: Information Systems Architectures** describes the development of Information Systems Architectures to support the agreed Architecture Vision
- **Phase D: Technology Architecture** describes the development of the Technology Architecture to support the agreed Architecture Vision
- **Phase E: Opportunities & Solutions** conducts initial implementation planning and the identification of delivery vehicles for the architecture defined in the previous phases
- **Phase F: Migration Planning** addresses how to move from the Baseline to the Target Architectures by finalizing a detailed Implementation and Migration Plan
- **Phase G: Implementation Governance** provides an architectural oversight of the implementation
- **Phase H: Architecture Change Management** establishes procedures for managing change to the new architecture
- **Requirements Management** operates the process of managing architecture requirements throughout the ADM

The description of the phases of the ADM in the TOGAF Standard — Architecture Development Method focuses on recommendations on *what* may be done to define and deploy an Enterprise Architecture.

Guidance on *how* to do what is specified can be found in the TOGAF Series Guides (see [Section 2.2](#)). A full list of referenced TOGAF Series Guides is included in [Appendix A](#).

The TOGAF framework recommends that the ADM be adapted to meet the needs of the enterprise and to support different architecture styles (see [Section 3.16](#)).

In particular, the ADM does not:

- Mandate that the phases must be performed in any specific sequence
- Mandate a "waterfall" method

The TOGAF Standard describes how the ADM can be used iteratively to develop a comprehensive Enterprise Architecture landscape. Rather than viewing the ADM graphic as a process model, it is helpful to view it as a reference model defining what has to be done in order to deliver solutions in an architected way and identifying interacting components across the enterprise and the relationships between them.

3.5 Enterprise Architecture Services

Activities described in the ADM are often provided through a service delivery model. The services are organized and presented in service categories. These services address specific needs independent of an organization's specific operation model. Any given service described utilizes the appropriate activities in the ADM to address a given need.

[Table 3-1](#) summarizes the proposed service categories and provides some context. The first four categories are customer-centric and the others are more internally centered on architects. Each service category is briefly described in the following subsections.

Table 3-1 Service Categories and Descriptors

Categories	Descriptor			
	Typical Customer	Typical Provider	Deliverable(s)	Desired Result
Customer-centric				
Enterprise Support Services	C-level management	Enterprise analysts using Enterprise Architecture as a tool	Answers to questions Assessment reports Recommendations	Better enterprise decisions Lower risk
Design Support Services	Program-level decision-makers	Enterprise Architect builder/modeler	MVAs (including standards and compliance criteria, roadmaps) for programs Compliance guidance Compliance reports	Better design decisions Successful programs and projects
Development Support Services	Project-level decision-makers	Enterprise Architect builder/modeler	MVAs (including standards and compliance criteria) for projects/products Compliance guidance Compliance reports	Better product decisions Successful products
Requirements Elicitation and Understanding Services	Product managers	Enterprise Architect with requirements understanding specialty	Stakeholder concerns Requirements Assessments (value, ability, etc.)	Solid outside-in view of requirements and value for solutions balanced among stakeholders
Internal-centric				
Architecture Planning Services	Architecture team leaders	Experienced Enterprise Architect	Architecture project plans	Resourced architecture team

Categories	Descriptor			
	Typical Customer	Typical Provider	Deliverable(s)	Desired Result
Enterprise Architecture Practice Development Support Services	Architecture organization decision-makers	Enterprise Architecture practice experts	Enterprise Architecture Capability assessments Enterprise Architecture Capability improvement recommendations	Highly skilled and organized Enterprise Architecture practice organization (internal or external)

3.5.1 Enterprise Support Services

This service category contains candidate services that enable informed enterprise decisions in support of organization change. These services could be provided independent of any individual project. These services focus on answering questions and providing enterprise analysis in support of strategic decisions.

3.5.2 Design Support Services

This service category contains candidate services that enable informed design decisions in support of organization change. These services would typically be provided after a project has been funded, whether large or small, waterfall or agile. These services include the development of Minimum Viable Architectures (MVAs) and associated analysis to support the design decisions.

3.5.3 Development Support Services

This service category contains candidate services that enable informed development decisions in support of organization change. These services would typically be provided during the development phase of a project, whether large or small, waterfall or agile. These services focus on answering questions and providing enterprise analysis in support of development decisions.

3.5.4 Requirements Elicitation and Understanding Services

This service category contains candidate services that enable requirements understanding. Taking a step beyond requirements management, these services help get closer to real need that will deliver greater business value.

3.5.5 Architecture Planning Services

This service category contains candidate services that enable well-planned and executed architecture projects in support of organization change. These services would typically be provided at the beginning of a "project" whether large or small, waterfall or agile.

3.5.6 Enterprise Architecture Practice Development Support Services

This service category contains candidate services that enable the development and management of an Enterprise Architecture practice. These services are focused on improving Enterprise Architecture Capability.

3.6 Deliverables, Artifacts, and Building Blocks

Architects executing the ADM will produce a number of outputs as a result of their efforts, such as process flows, architectural requirements, project plans, project compliance assessments, etc. The TOGAF Architecture Content Framework (see the TOGAF Standard — Architecture Content) provides a structural model for architectural content that allows major work products to be consistently defined, structured, and presented.

The Architecture Content Framework uses the following three categories to describe the type of architectural work product within the context of use:

- A **deliverable** is a work product that is contractually specified and in turn formally reviewed, approved, and signed off by the stakeholders

Deliverables represent the output of projects and those deliverables that are in documentation form will typically be archived at completion of a project, or transitioned into an Architecture Repository as a reference model, standard, or snapshot of the Architecture Landscape at a point in time.

- An **artifact** is an architectural work product that describes an aspect of the architecture

Artifacts are generally classified as catalogs (lists of things), matrices (showing relationships between things), and diagrams (pictures of things). Examples include a requirements catalog, application interaction matrix, and a value chain diagram. An architectural deliverable may contain one or more artifacts and artifacts will form the content of the Architecture Repository. An artifact may or may not be considered a deliverable based on the contractual specification.

- A **building block** represents a potentially re-usable component that can be combined with other building blocks to deliver architectures and solutions

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

- **Architecture Building Blocks (ABBs)** typically describe required capability and shape the specification of Solution Building Blocks (SBBs); for example, a customer services capability may be required within an enterprise, supported by many SBBs, such as processes, data, and application software

- **Solution Building Blocks (SBBs)** represent components that will be used to implement the required capability; for example, a network is a building block that can be described through complementary artifacts and then put to use to realize solutions for the enterprise

The relationships between deliverables, artifacts, and building blocks are shown in [Figure 3-2](#).

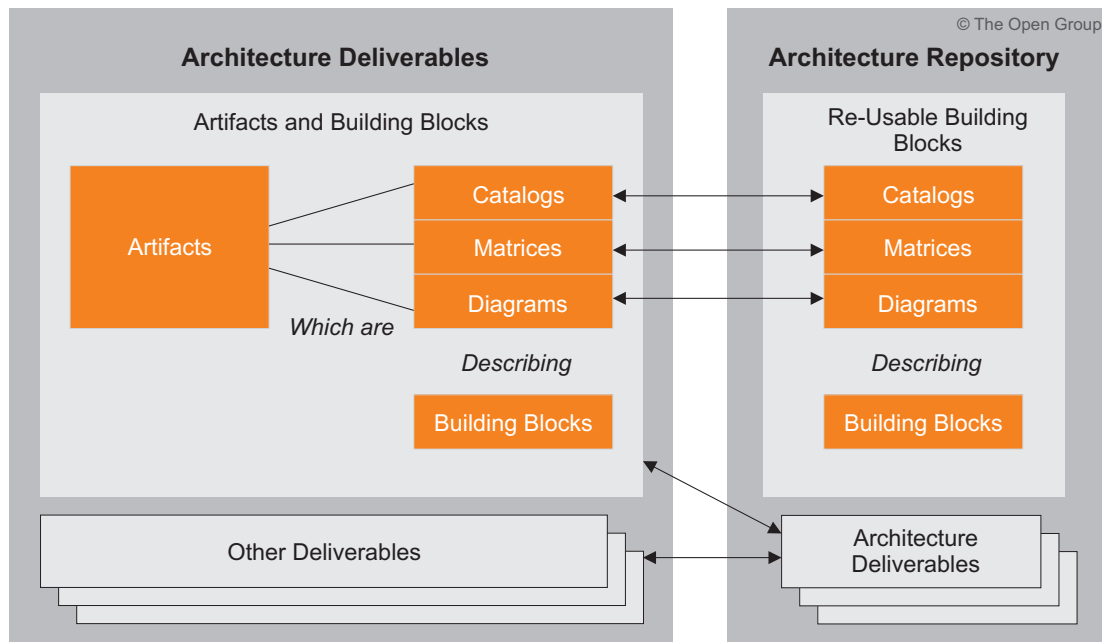


Figure 3-2 Relationships between Deliverables, Artifacts, and Building Blocks

For example, an Architecture Definition Document is a deliverable that documents an Architecture Description. This document will contain a number of complementary artifacts that are views of the building blocks relevant to the architecture. For example, a process flow diagram (an artifact) may be created to describe the target call handling process (a building block). This artifact may also describe other building blocks, such as the actors involved in the process (e.g., a Customer Services Representative). An example of the relationships between deliverables, artifacts, and building blocks is illustrated in [Figure 3-3](#).

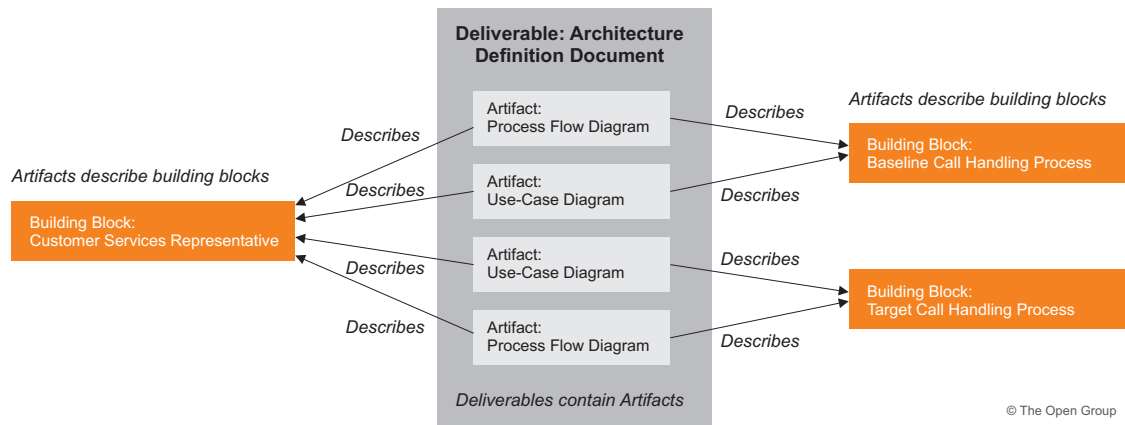


Figure 3-3 Example — Architecture Definition Document

The concepts of Deliverables, Artifacts, and Building Blocks are described in more detail in the TOGAF Standard — Architecture Content.

The TOGAF Standard — ADM Techniques describes the Architecture Development Method and includes summary lists of Deliverables and Artifacts that may be created in each phase. The TOGAF Standard — Architecture Content contains detailed descriptions of these.

3.7 Architecture Abstraction

An architectural technique for dividing a problem area into smaller problem areas that are easier to model and therefore easier to solve.

Abstraction levels are layered in nature, moving from high-level models to more detailed models.

Architecture effort can be divided into four distinct abstraction levels that cross the Business, Data, Application, and Technology domains to answer fundamental questions about an architecture:

- Why — why is the architecture needed?
- What — what functionality and other requirements need to be met by the architecture?
- How — how do we structure the functionality?
- With what — with what assets shall we implement this structure?

Note that why, what, and how have no connection to their use in the Zachman[®] Enterprise Architecture Framework.

3.7.1 Contextual Abstraction Level

This abstraction level is focused on understanding the environment in which an enterprise operates and the context in which architecture work is planned and executed. It answers why an enterprise undertakes architecture work, what is the scope of work, and the motivation in terms of goals, drivers, and objectives.

3.7.2 Conceptual Abstraction Level

This abstraction level is centered on decomposing the requirements to understand the problem, and what is needed to address the problem, without unduly focusing on how the architecture will be realized. It answers what is necessary to realize the requirements and is usually modeled using service models (business service, application service, technology service) that represent desired behavior.

Note this abstraction level can also be referred to as either service abstraction or behavior abstraction.

3.7.3 Logical Abstraction Level

This abstraction level is focused on identifying the kinds of business, data, application, and technology components needed to achieve the services identified in the conceptual level. It is about identifying how an architecture can be organized and structured, in an implementation-independent fashion. There will potentially be several ways to group services into logical components, based on principles and other grouping criteria, providing different logical solution alternatives.

3.7.4 Physical Abstraction Level

This abstraction level manages the allocation and implementation of physical components to meet the identified logical components. It is about determining with what physical components the logical-level components can be realized. There will potentially be many ways to use physical components to realize logical components, based on principles and other grouping criteria, providing different physical solution alternatives.

3.8 Architecture Principles

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

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

- **Enterprise Principles** provide a basis for decision-making throughout an enterprise, and inform how the organization sets about fulfilling its mission

Such principles are commonly found as a means of harmonizing decision-making across an organization. In particular, they are a key element in a successful Architecture Governance strategy (see TOGAF Standard — Enterprise Architecture Capability and Governance).

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

- **Architecture Principles** are a set of principles that relate to architecture work

They reflect a level of consensus across the enterprise and embody the spirit and thinking of existing Enterprise Principles. Architecture Principles govern the architecture process, affecting the development, maintenance, and use of the Enterprise Architecture.

Within an enterprise the hierarchy of principles starts with the Enterprise Principles. The subsidiary segment principles must exist within the bounds of these Enterprise Principles which are overarching. Consequently, at each hierarchical level the set of principles will be informed by and elaborate on the principles inherited from the level above and cannot overstep their boundaries.

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

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

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

Architecture Principles are further explained in the TOGAF Standard — ADM Techniques.

3.9 Interoperability

A definition of interoperability is "the ability to share information and services". Defining the degree to which the information and services are or are not to be shared is a very useful architectural requirement, especially in a complex organization and/or extended enterprise.

The determination of interoperability is present throughout the Architecture Development Method (ADM) as follows:

- In the Architecture Vision (Phase A), the nature and security considerations of the information and service exchanges are first revealed within the business scenarios
- In the Business Architecture (Phase B), the information and service exchanges are further defined in business terms
- In the Data Architecture (Phase C), the content of the information exchanges is detailed using the corporate data and/or information exchange model
- In the Application Architecture (Phase C), the way that the various applications are to share the information and services is specified
- In the Technology Architecture (Phase D), the appropriate technical mechanisms to permit the information and service exchanges are specified

- In Opportunities & Solutions (Phase E), the actual solutions (e.g., Commercial Off-The-Shelf (COTS) packages) are selected
- In Migration Planning (Phase F), the interoperability is logically implemented

There are many ways to define interoperability and the aim is to define one that is consistently applied within the enterprise and extended enterprise. It is best that both the enterprise and the extended enterprise use the same definitions.

Many organizations find it useful to categorize interoperability as follows:

- **Operational or Business Interoperability** defines how different parts of the enterprise work together at the business level
- **Information Interoperability** defines how information is to be shared
- **Technical Interoperability** defines how technical resources are to be shared or at least connect to one another

From an IT perspective, it is also useful to consider interoperability in a similar vein to Enterprise Application Integration (EAI); specifically:

- **Presentation Integration/Interoperability** is where a common look-and-feel approach through a common portal-like solution guides the user to the underlying functionality of the set of systems
- **Information Integration/Interoperability** is where the corporate information is seamlessly shared between the various corporate applications to achieve, for example, a common set of client information

Normally this is based upon a commonly accepted corporate ontology and shared services for the structure, quality, access, and security/privacy for the information.

- **Application Integration/Interoperability** is where the corporate functionality is integrated and shareable so that the applications are not duplicated (e.g., one change of address service/component; not one for every application) and are seamlessly linked together through functionality such as workflow

This impacts the business and infrastructure applications and is very closely linked to corporate business process unification/interoperability.

- **Technical Integration/Interoperability** includes common methods and shared services for the communication, storage, processing, and access to data primarily in the application platform and communications infrastructure domains

Interoperability and Interoperability Requirements are addressed in detail in the TOGAF Standard — ADM Techniques.

3.10 Enterprise Continuum

The TOGAF Standard includes the concept of the Enterprise Continuum, which sets the broader context for an architect and explains how generic solutions can be leveraged and specialized in order to support the requirements of an individual organization.

The Enterprise Continuum is a categorization for assets held in the Enterprise Repositories that provides methods for classifying assets, including architecture and solution artifacts as they evolve from generic Foundation Architectures to Organization-Specific Architectures. The

Enterprise Continuum comprises two complementary concepts: the Architecture Continuum and the Solutions Continuum.

The Enterprise Continuum is described in detail in the TOGAF Standard — Architecture Content.

An overview of the structure and context for the Enterprise Continuum is shown in [Figure 3-4](#).

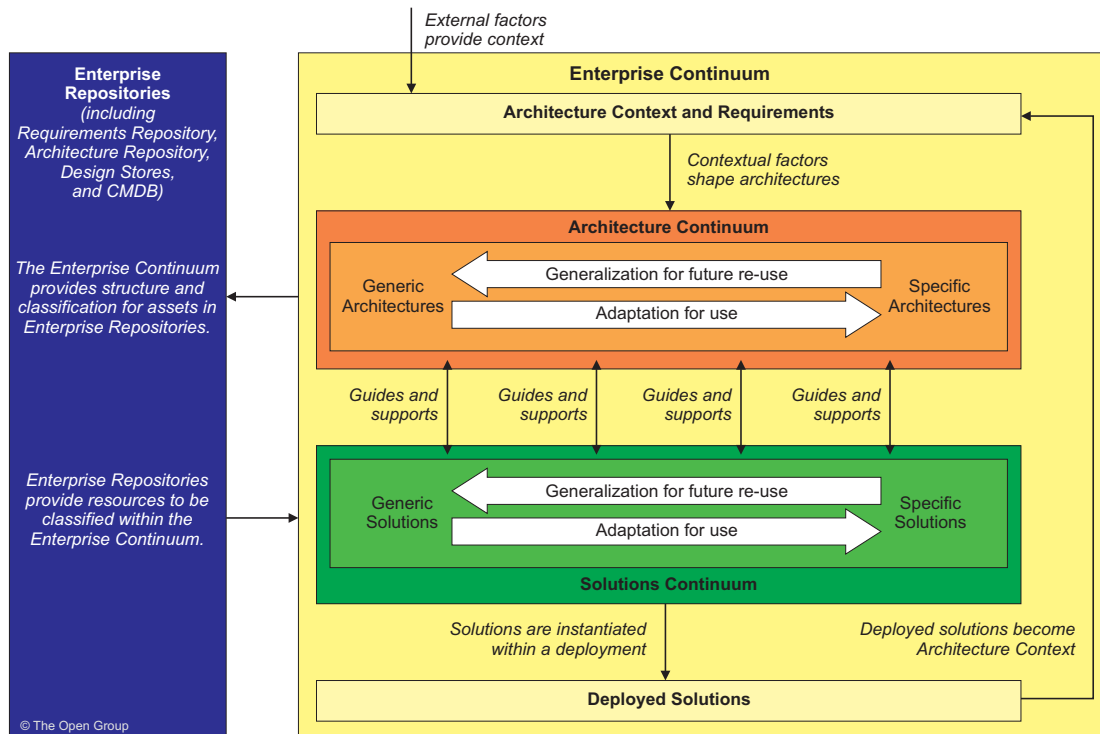


Figure 3-4 Enterprise Continuum

3.11 Architecture Repository

Supporting the Enterprise Continuum is the concept of an Architecture Repository which can be used to store different classes of architectural output at different levels of abstraction, created by the ADM. In this way, the TOGAF Standard facilitates understanding and co-operation between stakeholders and practitioners at different levels.

By means of the Enterprise Continuum and Architecture Repository, architects are encouraged to leverage all other relevant architectural resources and assets in developing an Organization-Specific Architecture. In this context, the TOGAF ADM can be regarded as describing a process lifecycle that operates at multiple levels within the organization, operating within a holistic governance framework and producing aligned outputs that reside in an Architecture Repository. The Enterprise Continuum provides a valuable context for understanding architectural models: it shows building blocks and their relationships to each other, and the constraints and requirements on a cycle of architecture development.

The structure of the TOGAF Architecture Repository is shown in Figure 3-5.

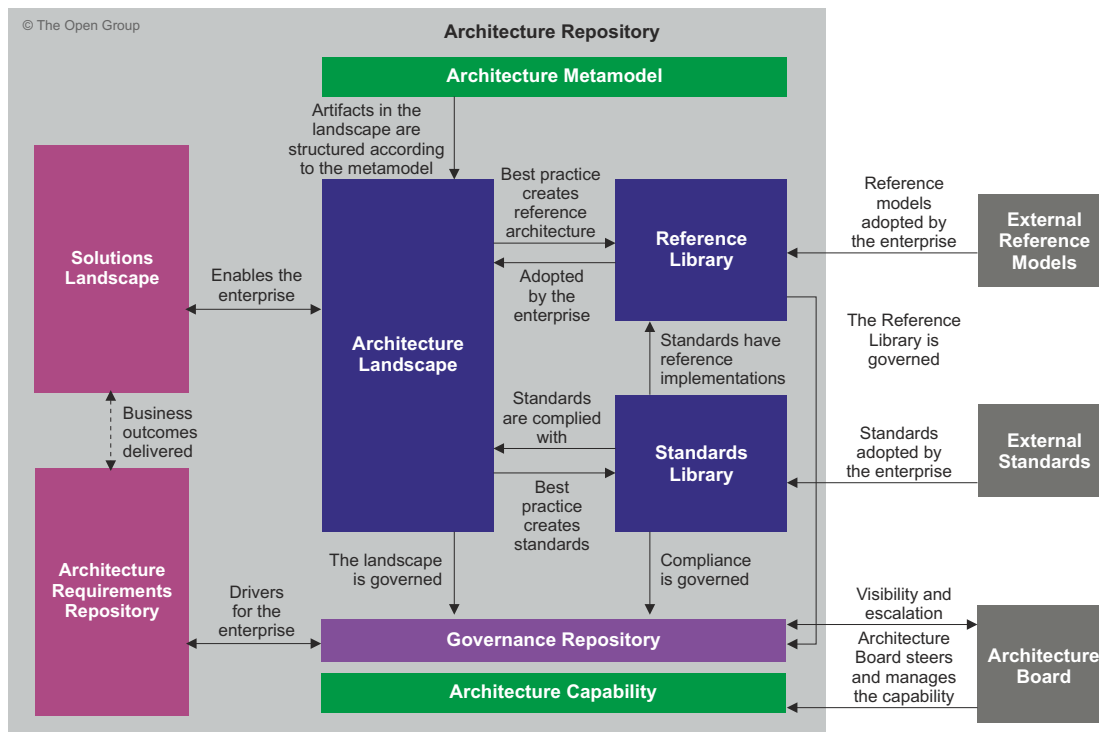


Figure 3-5 TOGAF Architecture Repository Structure

The major components within an Architecture Repository are as follows:

- The **Architecture Metamodel** describes the organizationally tailored application of an architecture framework, including a metamodel for architecture content
- The **Architecture Capability** defines the parameters, structures, and processes that support governance of the Architecture Repository
- The **Architecture Landscape** is the architectural representation of assets deployed within the operating enterprise at a particular point in time — the landscape is likely to exist at multiple levels of abstraction to suit different architecture objectives
- The **Standards Library** 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 Repository** provides a record of governance activity across the enterprise

- The **Architecture Requirements Repository** provides a view of all authorized architecture requirements which have been agreed with the Architecture Board
- The **Solutions Landscape** presents an architectural representation of the SBBs supporting the Architecture Landscape which have been planned or deployed by the enterprise

The TOGAF Architecture Repository is described in the TOGAF Standard — Architecture Content.

3.12 TOGAF Content Framework and Enterprise Metamodel

3.12.1 Overview

The TOGAF ADM provides lifecycle management to create and manage architectures within an enterprise. At each phase within the ADM, a discussion of inputs, outputs, and steps describes a number of architectural work products.

An essential task when establishing the enterprise-specific Enterprise Architecture Capability in the Preliminary Phase of the ADM is to define:

- A categorization framework to be used to structure the Architecture Descriptions, the work products used to express an architecture, and the collection of models that describe the architecture; this is referred to as the **Content Framework**
- An understanding of the types of entities within the enterprise and the relationships between them that need to be captured, stored, and analyzed in order to create the Architecture Description; this **Enterprise Metamodel** depicts this information in the form of a formal model
- The specific artifacts to be developed (see [Section 3.6](#))

The Content Framework chosen is likely to be influenced by:

- The Architecture Framework selected as the basis for the Enterprise Architecture Capability
- The chosen software tool used to support the Enterprise Architecture Capability

3.12.2 Content Framework

The Content Framework defines a categorization framework to be used to describe the building blocks and artifacts reflecting decisions taken in creating the overall architecture deliverables.

The Architecture Repository, which is explained in [Section 3.11](#), is structured to store the artifacts and work products identified in the Content Framework. The Content Framework is one element of the Enterprise-Specific Architecture Framework.

There are many alternative Content Frameworks (e.g., the TOGAF Content Framework, the Zachman Framework, DoDAF, NAF, etc.). Selecting a Content Framework is essential even though the choice of Content Framework is less important. The final Content Framework is usually adapted to fit specific organization needs.

The TOGAF Content Framework is intended to:

- Provide a detailed model of architectural work products
- Drive consistency in the outputs created when following the ADM
- Provide a comprehensive checklist of architecture output that could be created
- Reduce the risk of gaps within the final architecture deliverable set
- Help an enterprise mandate standard architecture concepts, terms, and deliverables

At the highest level, the TOGAF Content Framework (see [Figure 3-6](#)) is structured in line with the phases of the ADM.

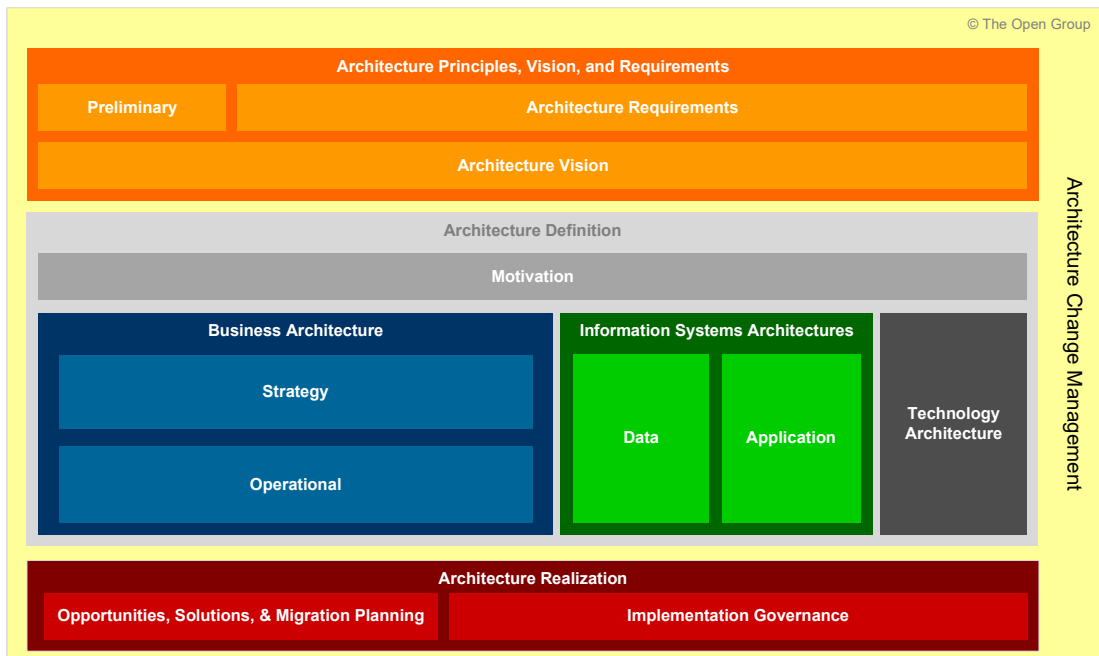


Figure 3-6 Content Framework by ADM Phase

- **Architecture Principles, Vision, Motivation, and Requirements** models are intended to capture the surrounding context of formal architecture models, including general Architecture Principles, strategic context that forms input for architecture modeling, and requirements generated from the architecture

The relevant aspects of the business context that have given rise to the Request for Architecture work are typically investigated, refined, validated, and recorded in the Preliminary and Architecture Vision phases.

- **Business Architecture** captures architecture models of the business, looking specifically at factors that motivate the enterprise, its structure, and its capabilities
- **Information Systems Architecture** models capture architecture models of IT systems, looking at applications and data in line with the TOGAF ADM phases

- **Technology Architecture** models capture technology assets that are used to implement and realize information system solutions
- **Architecture Realization/Transformation** models capture change roadmaps showing transition between architecture states and binding statements that are used to steer and govern an implementation of the architecture
- **Architecture Change Management** models capture value realization management events, internal and external, that impact the Enterprise Architecture and the generation of requirements for action

The TOGAF Content Framework is described in detail in the TOGAF Standard — Architecture Content.

3.12.3 Enterprise Metamodel

The TOGAF Standard encourages development of an Enterprise Metamodel, which defines the types of entity to appear in the models that describe the enterprise, together with the relationships between these entities.

For example, one type in an Enterprise Metamodel might be Role. Then the enterprise's Business Architecture models might include such instances of Role as Teller, Pilot, Manager, Volunteer, Customer, or Firefighter. Of course it would be an unusual enterprise that had all of these roles.

An Enterprise Metamodel provides value in several ways:

- It gives architects a starter set of the types of thing to investigate and to cover in their models
- It provides a form of completeness-check for any architecture modeling language, or architecture metamodel, that is proposed for use in an enterprise

Namely, how completely does it handle the types of entity in the Enterprise Metamodel, and manage required facts about them such as their attributes and relationships?

- It can help ensure:
 - Consistency
 - Completeness
 - Traceability

Note that the TOGAF Standard does not aim to constrain an enterprise's:

- Selection of artifacts
- Modeling notation

The TOGAF Standard may use a variety of modeling languages, such as the ArchiMate® modeling language, Business Process Modeling Notation™ (BPMN™), Unified Modeling Language™ (UML®), entity relationship diagramming, flowcharting, or any other notation that can express some TOGAF ideas.

The types of entity within an enterprise and the relationships between them are specific to the individual enterprise. Developing a high-quality metamodel is an important aspect of establishing the Enterprise Architecture Capability.

3.12.4 Developing the Enterprise Metamodel

The Enterprise Metamodel is an important part of the Organization-Specific Architecture Framework, as highlighted here. [Figure 3-7](#) shows how the Enterprise Continuum (see [Section 3.10](#)) provides a way to consider resources on a scale ranging from the most general ("Foundation") to most specific ("Organization-Specific"):

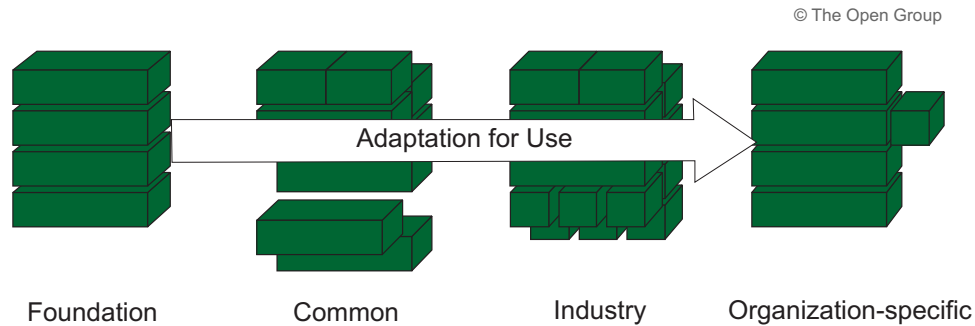


Figure 3-7 Applying the Enterprise Continuum

To support development of the enterprise's metamodel, the TOGAF Library includes a Foundation-level Core Enterprise Metamodel, detailed in the TOGAF Standard — Architecture Content. It shows types of entity, and relationships between them, that are likely to be required in modeling most enterprises and provides a context for the artifacts suggested in the ADM.

[illegible]

Figure 3-8 The TOGAF Core Enterprise Metamodel

3.13 Establishing and Maintaining an Enterprise Architecture Capability

In order to carry out architectural activity effectively within an enterprise, it is necessary to put in place an appropriate business capability for architecture, through organization structures, roles, responsibilities, skills, and processes. An overview of the TOGAF Architecture Capability is shown in [Figure 3-9](#).

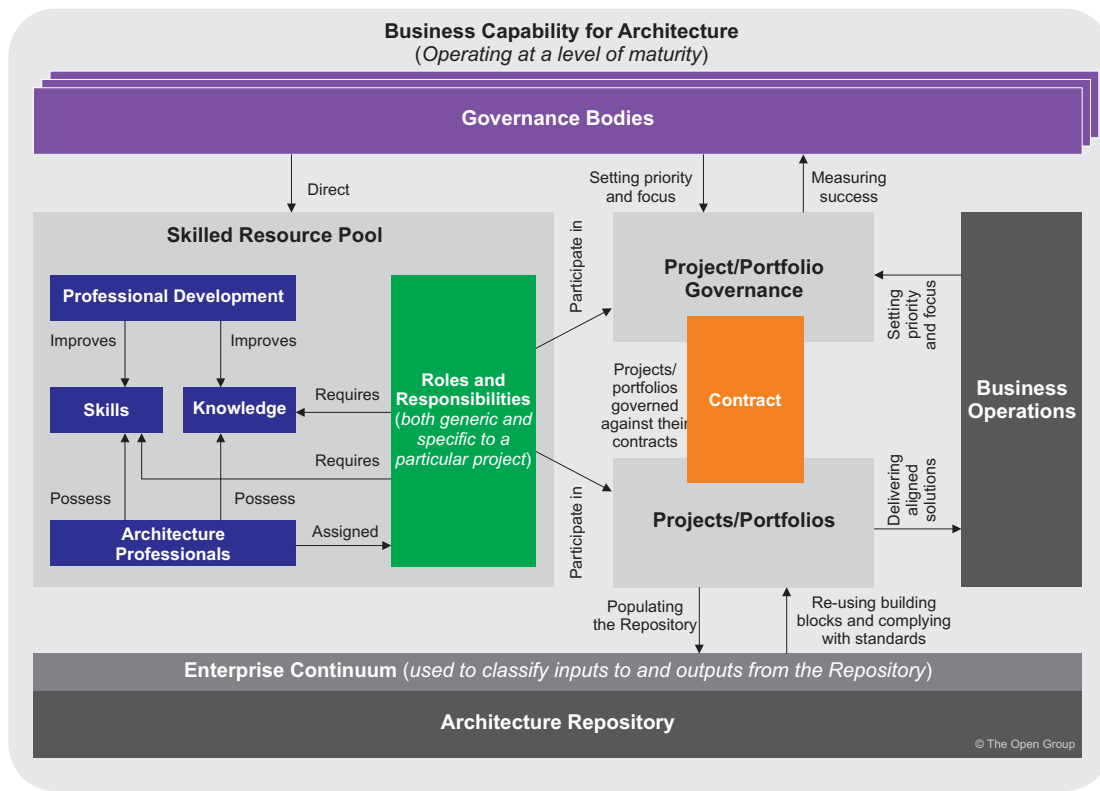


Figure 3-9 TOGAF Architecture Capability Overview

3.14 Establishing the Architecture Capability as an Operational Entity

Barring Architecture Capabilities set up to purely support change delivery programs, it is increasingly recognized that a successful Enterprise Architecture practice must sit on a firm operational footing. In effect, an Enterprise Architecture practice must be run like any other operational unit within a business; i.e., it should be treated like a business. To this end, and over and above the core processes defined within the ADM, an Enterprise Architecture practice should establish capabilities in the following areas:

- Financial Management
- Performance Management
- Service Management
- Risk and Opportunity Management (see [Section B.34](#))
- Resource Management
- Communications and Stakeholder Management (see [Section 4.36](#))

- Quality Management
- Supplier Management (see [Section B.40](#))
- Configuration Management (see [Section B.7](#))
- Environment Management

Central to the notion of operating an ongoing architecture is the execution of well-defined and effective governance, whereby all architecturally significant activity is controlled and aligned within a single framework.

As governance has become an increasingly visible requirement for organizational management, the inclusion of governance within the TOGAF Standard aligns the framework with current business best practice and also ensures a level of visibility, guidance, and control that will support all architecture stakeholder requirements and obligations.

The benefits of Architecture Governance include:

- Increased transparency of accountability, and informed delegation of authority
- Proactive risk and opportunity management
- Protection of the existing asset base through maximizing re-use of existing architectural components
- Proactive control, monitoring, and management mechanisms
- Process, concept, and component re-use across all organizational business units
- Value creation through monitoring, measuring, evaluation, and feedback
- Increased visibility supporting internal processes and external parties' requirements; in particular, increased visibility of decision-making at lower levels ensures oversight at an appropriate level within the enterprise of decisions that may have far-reaching strategic consequences for the organization
- Greater shareholder value; in particular, Enterprise Architecture increasingly represents the core intellectual property of the enterprise — studies have demonstrated a correlation between increased shareholder value and well-governed enterprises
- Integrates with existing processes and methodologies and complements functionality by adding control capabilities

Further detail on establishing an Enterprise Architecture Capability is given in the TOGAF Standard — Enterprise Architecture Capability and Governance.

3.15 Using the TOGAF Standard with Other Frameworks

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

With some exceptions, the majority of Enterprise Architecture frameworks focus on the first of these — the specific set of deliverables — and are relatively silent about the methods to be used to generate them (intentionally so, in some cases).

Because the TOGAF Standard is a generic framework and intended to be used in a wide variety

of environments, it provides a flexible and extensible content framework that underpins a set of generic architecture deliverables.

As a result, the TOGAF framework may be used either in its own right, with the generic deliverables that it describes; or else these deliverables may be replaced or extended by a more specific set, defined in any other framework that the architect considers relevant.

In all cases, it is expected that the architect will adapt and build on the TOGAF framework in order to define a tailored method that is integrated into the processes and organization structures of the enterprise. This architecture tailoring may include adopting elements from other architecture frameworks, or integrating TOGAF methods with other standard frameworks or best practices, such as ITIL®, CMMI®, COBIT®, PRINCE2, PMBOK, and MSP®. It may also include adopting reference materials from the TOGAF Library, such as the IT4IT™ Reference Architecture. Guidelines for adapting the TOGAF ADM in such a way are given in the TOGAF Standard — ADM Techniques.

As a generic framework and method for Enterprise Architecture, the TOGAF Standard provides the capability and the collaborative environment to integrate with other frameworks. Organizations are able to fully utilize vertical business domains, horizontal technology areas (such as security or manageability), or application areas (such as e-Commerce) to produce a competitive Enterprise Architecture framework which maximizes their business opportunities.

3.16 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 Framework
- 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.

3.17 Architecture Views and Viewpoints

The ability to create specific "views" of parts of a complex architecture is fundamental in being able to communicate with and allay concerns of stakeholders or groups of stakeholders. To gain full understanding and support from stakeholders it is necessary to present information in a form that each stakeholder will relate to and understand.

The role of architecture views is shown in [Figure 3-10](#), adapted from more formal definitions contained in ISO/IEC/IEEE 42010:2011 and ISO/IEC/IEEE 15288:2015.

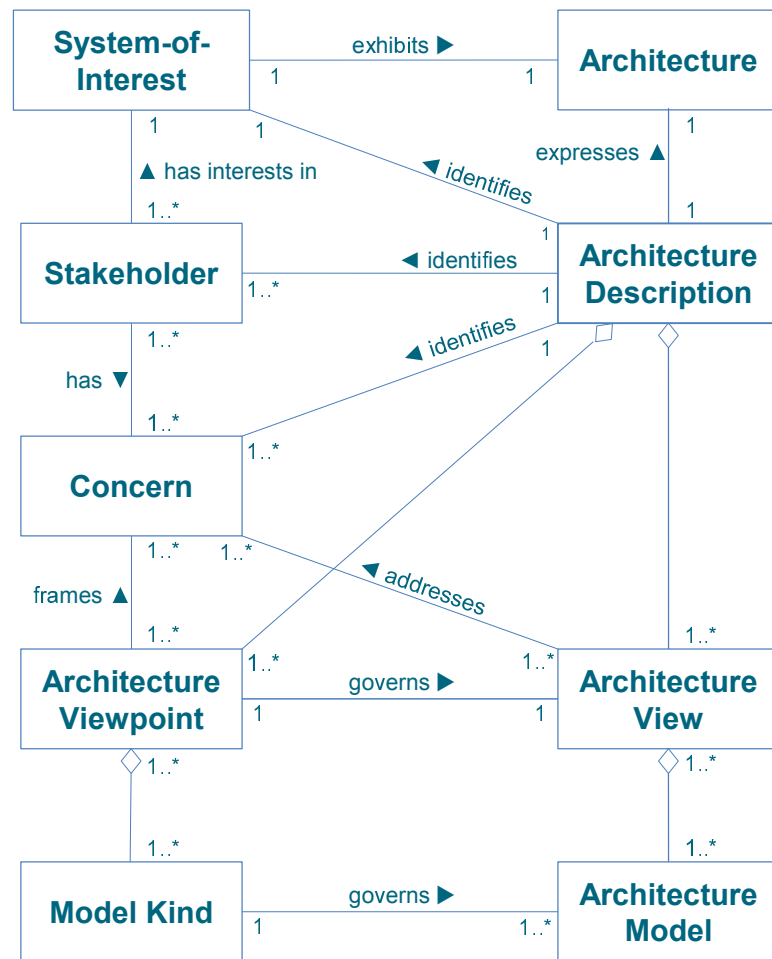


Figure 3-10 Basic Architectural Concepts

3.18 Enterprise Agility

Enterprise agility is a commonly used term, but the exact definition differs among practitioners. Regardless of how the term is defined, it is important because it enables an enterprise to better react to change by being more customer and product-centric, more efficient, and better able to ensure regulatory compliance.

The term "agile" is frequently associated with agile software development processes associated with the Manifesto for Agile Software Development.

While these "agile" principles and techniques can be applied to adapt the TOGAF framework, enterprise agility is a broader context than agile software development. Therefore, additional techniques are employed in adapting the TOGAF framework to an agile enterprise.

Enterprise Architecture provides a framework for change, linked to both strategic direction and business value. It provides a sufficient view of the organization to manage complexity, support

continuous change, and manage the risk of unanticipated consequences.

The TOGAF framework has embraced the call to respond to the needs of the enterprise in a timely manner, through the concepts of "partitions" and "levels". Partitions define how the work is broken down into multiple architecture initiatives. Levels define how the overall architecture can be developed at different levels of granularity and detail.

In addition, the TOGAF ADM supports a number of concepts that are characterized as iteration.

More detailed descriptions of how to adapt the TOGAF ADM to support enterprise agility can be found in:

- TOGAF® Series Guide: Applying the ADM Using Agile Sprints
- TOGAF® Series Guide: Enabling Enterprise Agility
- The Open Agile Architecture™ Standard

3.19 Risk Management

There will always be risk with any architecture/business transformation effort. It is important to identify, classify, and mitigate these risks before starting so that they can be tracked throughout the transformation effort.

Mitigation is an ongoing effort and often the risk triggers may be outside the scope of the transformation planners (e.g., merger, acquisition) so planners must monitor the transformation context constantly.

It is also important to note that the Enterprise Architect may identify the risks and mitigate certain ones, but it is within the governance framework that risks have to be first accepted and then managed.

There are two levels of risk that should be considered, namely:

- Initial level of risk: risk categorization prior to determining and implementing mitigating actions
- Residual level of risk: risk categorization after implementation of mitigating actions (if any)

The process for risk management consists of the following activities:

- Risk classification
- Risk identification
- Initial risk assessment
- Risk mitigation and residual risk assessment
- Risk monitoring

A qualitative approach to risk management is described in the TOGAF Standard — ADM Techniques.

Risk concepts are included in the Enterprise Security Architecture described in the TOGAF® Series Guide: Integrating Risk and Security within a TOGAF® Enterprise Architecture.

A more rigorous quantitative approach is described in the Open FAIR™ Body of Knowledge which comprises two standards from The Open Group: Open Risk Taxonomy (O-RT) and Open Risk Analysis (O-RA).

Chapter 4: Definitions

For the purposes of the TOGAF Standard, the following terms and definitions apply. [Appendix B](#) should be referenced for supplementary definitions not defined in this chapter. The Merriam-Webster® Collegiate Dictionary should be referenced for terms not defined in this section or [Appendix B](#).

4.1 Abstraction

The technique of providing summarized or generalized descriptions of detailed and complex content.

Note: Abstraction, as in "level of abstraction", can also mean providing a focus for analysis that is concerned with a consistent and common level of detail or abstraction. Abstraction in this sense is typically used in architecture to allow a consistent level of definition and understanding to be achieved in each area of the architecture in order to support effective communication and decision-making. It is especially useful when dealing with large and complex architectures as it allows relevant issues to be identified before further detail is attempted.

4.2 Actor

A person, organization, or system that has one or more roles that initiates or interacts with activities; for example, a sales representative who travels to visit customers. Actors may be internal or external to an organization.

Note: In the automotive industry, an original equipment manufacturer would be considered an actor by an automotive dealership that interacts with its supply chain activities.

4.3 Application Architecture

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

Note: Application Architecture is described in the TOGAF Standard — Architecture Development Method.

4.4 Application Component

An encapsulation of application functionality aligned to implementation structure, which is modular and replaceable. It encapsulates its behavior and data, provides services, and makes them available through interfaces.

Note: For example, a business application such as an accounting, payroll, or CRM system.

An application component usually maintains a data component. It is enabled by technology services provided by technology components.

4.5 Application Platform

The collection of technology components of hardware and software that provide the services used to support applications.

4.6 Application Service

A discrete behavior requestable from an application; an automated element supporting or delivering part or all of one or more business services.

4.7 Architectural Style

The combination of distinctive features related to the specific context within which architecture is performed or expressed; a collection of principles and characteristics that steer or constrain how an architecture is formed.

4.8 Architecture

1. The fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution. (Source: ISO/IEC/IEEE 42010:2011)
2. The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.

4.9 Architecture Building Block (ABB)

An architectural component that specifies the required Solution Building Blocks (SBBs) at a more logical (or supplier-independent) level.

See also [Section 4.26](#).

4.10 Architecture Continuum

A categorization mechanism, with increasing detail and specialization, for the components and artifacts stored in the Architecture Landscape or Reference Library (part of the Architecture Repository).

Note: This Continuum begins with foundational definitions like reference models, core strategies, and basic building blocks. From there it spans to Industry Architectures and all the way to an Organization-Specific Architecture.

See also [Section 4.44](#).

4.11 Architecture Development Method (ADM)

The core of the TOGAF framework. A multi-phase, iterative approach to develop and use an Enterprise Architecture to shape and govern business transformation.

Note: The ADM is described in the TOGAF Standard — ADM Techniques.

4.12 Architecture Domain

The architectural area being considered. The TOGAF framework follows the tradition of dividing Enterprise Architecture into four primary architecture domains: business, data, application, and technology. Other domains (motivation, security, governance, etc.) may span those four primary domains.

4.13 Architecture Framework

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

4.14 Architecture Governance

The practice of monitoring and directing architecture-related work. The goal is to deliver desired outcomes and adhere to relevant principles, standards, and roadmaps.

See also [Section 4.48](#).

4.15 Architecture Landscape

The architectural representation of assets in use, or planned, by the enterprise at particular points in time.

4.16 Architecture Level

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

Note: Architecture levels are distinct from architecture partitions.

4.17 Architecture Model

A representation of a subject of interest.

Note: An architecture model provides a smaller scale, simplified, and/or abstract representation of the subject matter.

See also [Section 4.75](#), [Section 4.20](#), and [Section 4.21](#).

4.18 Architecture Partition

A subset of architecture resulting from dividing that architecture to facilitate its development and management.

4.19 Architecture Principle

A qualitative statement of intent that should be met by the architecture.

Note: A sample set of Architecture Principles is defined in the TOGAF Standard — ADM Techniques.

4.20 Architecture View

A representation of a system from the perspective of a related set of concerns.

Note: In some sections of this standard, the term "view" is used as a synonym for "architecture view".

See also [Section 4.75](#) and [Section 4.21](#).

4.21 Architecture Viewpoint

A specification of the conventions for a particular kind of architecture view.

Note: An architecture viewpoint can also be seen as the definition or schema for that kind of architecture view. It establishes the conventions for constructing, interpreting, and using an architecture view to address a specific concern (or set of concerns) about a system-of-interest.

In some sections of this standard, the term "viewpoint" is used as a synonym for "architecture viewpoint".

See also [Section B.25](#).

4.22 Architecture Vision

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

Note: Phase A (Architecture Vision) is described in the TOGAF Standard — Architecture Development Method.

4.23 Artifact

An architectural work product that describes an aspect of the architecture.

See also [Section 4.26](#).

4.24 Baseline

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

4.25 Boundaryless Information Flow™

A shorthand representation of "access to integrated information to support business process improvements" representing a desired state of an enterprise's infrastructure specific to the business needs of the organization.

Note: The need for Boundaryless Information Flow — a trademark of The Open Group — is described in the TOGAF® Series Guide: The TOGAF Integrated Information Infrastructure Reference Model (III-RM).

4.26 Building Block

A potentially re-usable component that can be combined with other building blocks to deliver architectures and solutions.

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

Building blocks are described in the TOGAF Standard — Architecture Content.

See also [Section 4.23](#).

4.27 Business Architecture

A representation of holistic, multi-dimensional business views of: capabilities, end-to-end value delivery, information, and organizational structure; and the relationships among these business views and strategies, products, policies, processes, initiatives, and stakeholders.

Note: Business Architecture relates business elements to business goals and elements of other domains.

Business Architecture is described in the TOGAF Standard — Architecture Development Method.

4.28 Business Capability

A particular ability that a business may possess or exchange to achieve a specific purpose.

4.29 Business Function

A collection of business behavior based on a chosen set of criteria, closely aligned to an organization.

4.30 Business Governance

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

4.31 Business Model

A model describing the rationale for how an enterprise creates, delivers, and captures value.

4.32 Business Service

Supports the business by encapsulating a unique "element of business behavior".

Note: A service offered external to the enterprise may be supported by business services.

4.33 Capability

An ability that an organization, person, or system possesses.

Note: This a general-purpose definition. See [Section 4.28](#) for how this concept is refined for usage in Business Architecture.

4.34 Capability Architecture

An architecture that describes the abilities that an enterprise possesses.

See also the TOGAF® Series Guide: A Practitioners' Approach to Developing Enterprise Architecture Following the TOGAF ADM.

4.35 Capability Increment

A discrete portion of a capability architecture that delivers specific value. When all increments have been completed, the capability has been realized.

4.36 Communications and Stakeholder Management

The management of needs of stakeholders of the Enterprise Architecture practice. It also manages the execution of communication between the practice and the stakeholders and the practice and the consumers of its services.

Note: Architecture stakeholder management is described in the TOGAF Standard — ADM Techniques.

4.37 Concern

An interest in a system relevant to one or more of its stakeholders.

Note: Concerns may pertain to any aspect of the system's functioning, development, or operation, including considerations such as performance, reliability, security, distribution, and evolvability and may determine the acceptability of the system.

See also [Section 4.75](#).

4.38 Course of Action

Direction and focus provided by strategic goals and objectives, often to deliver the value proposition characterized in the business model.

4.39 Data Architecture

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

Note: Data Architecture is described in the TOGAF Standard — Architecture Development Method.

4.40 Deliverable

An architectural work product that is contractually specified and in turn formally reviewed, agreed, and signed off by the stakeholders.

Note: Deliverables represent the output of projects and those deliverables that are in documentation form will typically be archived at completion of a project, or transitioned into an Architecture Repository as a reference model, standard, or snapshot of the Architecture Landscape at a point in time.

4.41 Digital Architecture

The inclusive architecture focused on a combination of Enterprise Architecture, data science, telecommunications and IoT, security, artificial intelligence, cognitive science, neuroscience, robotics, and social medias to deliver operational services.

4.42 Enterprise

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

4.43 Enterprise Architecture Service

An encapsulated element of Enterprise Architecture capability that delivers specific Enterprise Architecture functionality.

4.44 Enterprise Continuum

A categorization mechanism for classifying architecture and Solution Building Blocks (SBBs) as they evolve from generic to specific applicability (or *vice versa*).

See also [Section 4.10](#) and [Section 4.74](#).

4.45 Foundation Architecture

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

4.46 Framework

A structure for content or process that can be used as a tool to structure thinking, ensuring consistency and completeness.

4.47 Gap

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

Note: Gap analysis is described in the TOGAF Standard — ADM Techniques.

4.48 Governance

The discipline of monitoring and guiding the management of a business (or IS/IT landscape) to deliver the business outcomes required.

See also [Section 4.14](#), [Section 4.30](#), and [Section B.27](#) in [Appendix B](#).

4.49 Information

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

4.50 Information Technology (IT)

The lifecycle management of information and related technology used by an organization.

4.51 Interoperability

1. The ability to share information and services.
2. The ability of two or more systems or components to exchange and use information.
3. The ability of systems to provide and receive services from other systems and to use the services so interchanged to enable them to operate effectively together.

4.52 Logical

Implementation-independent.

Note: A logical architecture is an implementation-independent definition of the architecture.

4.53 Metadata

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

4.54 Metamodel

A model that describes the entities used in building an Architecture Description, their characteristics, and the key relationships between those entities.

4.55 Method

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

4.56 Modeling

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

4.57 Model Kind

Conventions for a type of modeling.

Note: An architecture viewpoint references one or more model kinds; an architecture view incorporates one or more models.

4.58 Objective

An organizational aim that is declared in a Specific, Measurable, Actionable, Realistic, and Timebound (SMART) way. For example, "Increase capacity utilization by 30% by the end of the year, to support the planned increase in market share".

4.59 Pattern

A technique for putting building blocks into context; for example, to describe a re-usable solution to a problem.

Note: Building blocks are what you use: (architecture) patterns can tell you how you use them, when, why, and what trade-offs you have to make in doing so.

See also [Section 4.26](#).

4.60 Physical

Real-world, tangible.

Note: A logical architecture is realized through a physical architecture.

4.61 Principle

See [Section 4.19](#).

4.62 Product

An outcome generated by the business to be offered to customers. Products include materials and/or services.

4.63 Reference Model (RM)

An abstract framework for understanding significant relationships among the entities of [an] environment, and for the development of consistent standards or specifications supporting that environment.

Note: A reference model is based on a small number of unifying concepts and may be used as a basis for education and explaining standards to a non-specialist. A reference model is not directly tied to any standards, technologies, or other concrete implementation details, but it does seek to provide common semantics that can be used unambiguously across and between different implementations.

Source: OASIS®; refer to www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm.

4.64 Requirement

A statement of need, which is unambiguous, testable or measurable, and necessary for acceptability.

4.65 Roadmap

An abstracted plan for business or technology change, typically operating across multiple disciplines over multiple years. Normally used in the phrases Technology Roadmap, Architecture Roadmap, etc.

4.66 Role

1. The usual or expected behavior of an actor, or the part somebody or something plays in a particular process or event. An actor may have a number of roles.
2. The part an individual plays in an organization and the contribution they make through the application of their skills, knowledge, experience, and abilities.

See also [Section 4.2](#).

4.67 Segment Architecture

A detailed, formal description of areas within an enterprise, used at the program or portfolio level to organize and align change activity.

See also [Section 4.77](#).

4.68 Service

An encapsulated element of behavior that provides specific functionality in response to requests from actors or other services.

Note: A service has an interface and description.

4.69 Service Orientation

Viewing an enterprise, system, or building block in terms of services provided and consumed.

See also [Section 4.70](#).

4.70 Service-Oriented Architecture (SOA)

An architectural style that supports service orientation.

See also [Section 4.7](#) and [Section 4.69](#).

4.71 Service Portfolio

A collection of services, potentially an interface definition.

Note: It is used in the TOGAF framework to define the requirement for a building block or system.

4.72 Solution Architecture

A description of a discrete and focused business operation or activity and how IS/IT supports that operation.

4.73 Solution Building Block (SBB)

A physical or implementation-specific component that realizes part or all of one or more logical Architecture Building Blocks (ABBs).

Note: There are business, application, and technology SBBs.

4.74 Solutions Continuum

A categorization mechanism, with increasing detail and specialization, for the components and artifacts stored in the Solutions Landscape or Reference Library (part of the Architecture Repository).

See also [Section 4.44](#) and [Section 4.10](#).

4.75 Stakeholder

An individual, team, organization, or class thereof, having an interest in a system.

4.76 Standards Library

A library of standards that can be used to define the particular services and other components of an Organization-Specific Architecture.

Note: The Standards Library is described in the TOGAF Standard — Architecture Content: Architecture Repository.

4.77 Strategic Architecture

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

4.78 Target Architecture

The description of a future state of the architecture being developed for an organization.

Note: There may be several future states developed as a roadmap to show the evolution of the architecture to a target state.

4.79 Taxonomy of Architecture Views

The organized collection of all architecture views pertinent to an architecture.

4.80 Technology Architecture

A description of the structure and interaction of the technology services and technology components.

Note: Technology Architecture is described in the TOGAF Standard — Architecture Development Method.

4.81 Technology Component

1. A technology building block. A generic infrastructure technology that supports and enables application or data components (directly or indirectly) by providing technology services.
2. An encapsulation of technology infrastructure that represents a class of technology product or specific technology product.

4.82 Technology Service

A technical capability required to provide enabling infrastructure that supports the delivery of applications.

4.83 Transition Architecture

A formal description of one state of the architecture at an architecturally significant point in time.

Note: One or more Transition Architectures may be used to describe the progression in time from the Baseline to the Target Architecture.

Transition Architecture is described in the TOGAF Standard — Architecture Content: Architecture Definition Document.

4.84 Value Stream

A representation of an end-to-end collection of activities that create an overall result for a customer, stakeholder, or end user.

4.85 View

See [Section 4.20](#).

4.86 Viewpoint

See [Section 4.21](#).

4.87 Viewpoint Library

A collection of the specifications of architecture viewpoints contained in the Reference Library portion of the Architecture Repository.

4.88 Work Package

A set of actions identified to achieve one or more objectives for the business. A work package can be a part of a project, a complete project, or a program.

Appendix A: Referenced Documents

The Open Group TOGAF Series Guides

- TOGAF® Series Guide: Applying the ADM Using Agile Sprints, April 2022 (G210), published by The Open Group; refer to www.opengroup.org/library/g210
- TOGAF® Series Guide: A Practitioners' Approach to Developing Enterprise Architecture Following the TOGAF ADM, April 2022 (G186), published by The Open Group; refer to www.opengroup.org/library/g186
- TOGAF® Series Guide: Architecture Maturity Models, April 2022 (G203), published by The Open Group; refer to www.opengroup.org/library/g203
- TOGAF® Series Guide: Architecture Project Management, April 2022 (G188), published by The Open Group; refer to www.opengroup.org/library/g188
- TOGAF® Series Guide: Architecture Skills Framework, April 2022 (G198), published by The Open Group; refer to www.opengroup.org/library/g198
- TOGAF® Series Guide: Business Capabilities, Version 2, April 2022 (G211), published by The Open Group; refer to www.opengroup.org/library/g211
- TOGAF® Series Guide: Business Models, April 2022 (G18A), published by The Open Group; refer to www.opengroup.org/library/g18a
- TOGAF® Series Guide: Business Scenarios, April 2022 (G176), published by The Open Group; refer to www.opengroup.org/library/g176
- TOGAF® Series Guide: Digital Technology Adoption — A Guide to Readiness Assessment and Roadmap Development, April 2022 (G212), published by The Open Group; refer to www.opengroup.org/library/g212
- TOGAF® Series Guide: Information Architecture — Customer Master Data Management, April 2022 (G21B), published by The Open Group; refer to www.opengroup.org/library/g21b
- TOGAF® Series Guide: Information Mapping, April 2022 (G190), published by The Open Group; refer to www.opengroup.org/library/g190
- TOGAF® Series Guide: Integrating Risk and Security within a TOGAF® Enterprise Architecture, April 2022 (G152), published by The Open Group; refer to www.opengroup.org/library/g152
- TOGAF® Series Guide: Organization Mapping, April 2022 (G206), published by The Open Group; refer to www.opengroup.org/library/g206
- TOGAF® Series Guide: Enabling Enterprise Agility, April 2022 (G20F), published by The Open Group; refer to www.opengroup.org/library/g20f
- TOGAF® Series Guide: The TOGAF Integrated Information Infrastructure Reference Model (III-RM): An Architected Approach to Boundaryless Information Flow™, November 2017 (G179), published by The Open Group; refer to www.opengroup.org/library/g179

- TOGAF® Series Guide: The TOGAF Leader's Guide to Establishing and Evolving an EA Capability, April 2022 (G184), published by The Open Group; refer to www.opengroup.org/library/g184
- TOGAF® Series Guide: The TOGAF Technical Reference Model (TRM), September 2017 (G175), published by The Open Group; refer to www.opengroup.org/library/g175
- TOGAF® Series Guide: Using the TOGAF® Framework to Define and Govern Service-Oriented Architectures, September 2017 (G174), published by The Open Group; refer to www.opengroup.org/library/g174
- TOGAF® Series Guide: Value Streams, April 2022 (G178), published by The Open Group; refer to www.opengroup.org/library/g178

Other Publications from The Open Group

- A Practical Approach to Application Portfolio Consolidation using the TOGAF® Standard, The Open Group Guide, July 2018 (G18B), published by The Open Group; refer to www.opengroup.org/library/g18b
- Archi Banking Group: Combining the BIAN Reference Model, ArchiMate® Modeling Notation, and the TOGAF® Framework, Case Study, March 2020 (Y201), published by The Open Group; refer to www.opengroup.org/library/y201
- ArchiMate® 3.1 Specification, The Open Group Standard, November 2019 (C197), published by The Open Group; refer to www.opengroup.org/library/c197
- Digital Practitioner Body of Knowledge™ Standard (also known as the DPBoK™ Standard), The Open Group Standard, January 2020 (C196), published by The Open Group; refer to www.opengroup.org/library/c196
- Exploring Synergies between TOGAF® and Frameworkx, White Paper, May 2011 (W114), published by The Open Group; refer to www.opengroup.org/library/w114
- Information Architecture: Business Intelligence & Analytics and Metadata Management Reference Models, The Open Group Guide, January 2020 (G201), published by The Open Group; refer to www.opengroup.org/library/g201
- Open Risk Analysis (O-RA), Version 2.0, The Open Group Standard, November 2020 (C20A), published by The Open Group; refer to www.opengroup.org/library/c20a
- Open Risk Taxonomy (O-RT), Version 3.0, The Open Group Standard, November 2020 (C20B), published by The Open Group; refer to www.opengroup.org/library/c20b
- The Open Agile Architecture™ Standard, The Open Group Standard, September 2020 (C208), published by The Open Group; refer to www.opengroup.org/library/c208
- The Open Group IT4IT™ Reference Architecture, Version 2.1, The Open Group Standard, January 2017 (C171), published by The Open Group; refer to www.opengroup.org/library/c171
- TOGAF® 9 and DoDAF 2.0, White Paper, July 2010 (W105), published by The Open Group; refer to www.opengroup.org/library/w105
- TOGAF® and SABSA® Integration, White Paper, October 2011 (W117), published by The Open Group; refer to www.opengroup.org/library/w117

Other Referenced Documents

- A Guide to the Project Management Body of Knowledge (PMBOK®) Guide), Project Management Institute; refer to www.pmi.org/pmbok-guide-standards
- A Method for Identifying Process Re-Use Opportunities to Enhance the Operating Model, M. de Vries, A. van der Merwe, P. Kotze, A. Gerber, in proceedings of the 2011 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)
- Analysis Patterns — Reusable Object Models, M. Fowler, ISBN: 0-201-89542-0, Addison-Wesley
- ANSI/IEEE Std 1471-2000: Systems and Software Engineering — Recommended Practice for Architectural Description of Software-intensive Systems
- A Pattern Language: Towns, Buildings, Construction, Christopher Alexander, ISBN: 0-19-501919-9, Oxford University Press, 1979
- Business Motivation Model™ (BMM™), Object Management Group (OMG); refer to www.omg.org/spec/BMM/About-BMM
- Business Process Modeling Notation™ (BPMN™) Specification, Object Management Group (OMG); refer to www.bpmn.org
- Business Transformation Enablement Program (BTEP), Canadian Government; refer to <https://purl.org/theopengroup/btep>
- Control Objectives for Information and related Technology (COBIT®), Version 4.0, IT Governance Institute, 2005
- Corporate Governance, Ranami Naidoo, ISBN: 1-919-903-0086, Double Storey, 2002
- Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, ISBN: 0-201-63361-2, Addison-Wesley, October 1994
- Enterprise Architecture as Strategy, Jeanne Ross, Peter Weill, David C. Robertson, ISBN: 1-59139-839-8, Harvard Business School Press, 2006
- Enterprise Architecture Capability Maturity Model (ACMM), Version 1.2, United States Department of Commerce, December 2007
- Enterprise Architecture Maturity Model, Version 1.3, National Association of State CIOs (NASCIO), December 2003
- Enterprise Architecture Planning (EAP): Developing a Blueprint for Data, Applications, and Technology, Steven H. Spewak, Steven C. Hill, ISBN: 0-47-159985-9, John Wiley & Sons, 1993
- Headquarters Air Force Principles for Information Management, US Air Force, June 29, 1998
- ISO/IEC 20000:2011, Information Technology — Service Management
- ISO/IEC/IEEE 15288:2015, Systems and Software Engineering — System Life Cycle Processes
- ISO/IEC/IEEE 42010:2011, Systems and Software Engineering — Architecture Description
- IT Portfolio Management Facility™ (ITPMF™) Specification, Object Management Group (OMG); refer to www.omg.org/spec/ITPMF
- Manifesto for Agile Software Development, 2001; refer to www.agilemanifesto.org
- Merriam-Webster Collegiate Dictionary, Merriam-Webster, English Language, 11th Edition, April 2008, ISBN-10: 0877798095, ISBN-13: 978-0877798095; refer to www.merriam-webster.com

- Model-Driven Architecture® (MDA®) Specification, Object Management (OMG); refer to www.omg.org/mda
- OECD Principles of Corporate Governance, Organization for Economic Co-operation and Development, December 2001; refer to www.oecd.org
- Organigraphs: Drawing How Companies Really Work, H. Mintzberg and L. Van der Heyden, Harvard Business Review, October 1999; refer to <https://hbr.org/1999/09/organigraphs-drawing-how-companies-really-work>
- Pattern-Oriented Software Architecture: A System of Patterns, F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, ISBN: 0-471-95869-7, John Wiley & Sons, 1996
- Patterns and Software: Essential Concepts and Terminology, Brad Appleton, 2000; refer to www.bradapp.com/docs/patterns-intro.html
- Re-usable Asset Specification (RAS), Version 2.2, Object Management Group (OMG), November 2005; refer to www.omg.org/spec/RAS
- Service Component Architecture (SCA) Specification, developed by the Open Service Oriented Architecture (OSOA) collaboration; refer to www.oasis-opencsa.org/sca
- Service Data Objects (SDO) Specification, developed by the Open Service Oriented Architecture (OSOA) collaboration; refer to www.oasis-opencsa.org/sdo
- Software Processing Engineering Metamodel (SPEM™) Specification, Version 2.0, Object Management Group (OMG), April 2008; refer to www.omg.org/spec/SPEM
- Systems Modeling Language™ (SysML®), Object Management Group (OMG); refer to www.sysml.org
- The Art of Systems Architecting, Eberhardt Rechtin, Mark W. Maier, 2000
- The Oregon Experiment, Christopher Alexander, ISBN: 0-19-501824-9, Oxford University Press, 1975
- The Timeless Way of Building, Christopher Alexander, ISBN: 0-19-502402-8, Oxford University Press, 1979
- The Zachman® Framework, Zachman International; refer to www.zachman.com
- UML Profile and Metamodel for Services (UPMS) RFP (OMG soa/2006-09-09), Object Management Group (OMG), June 2007
- Unified Modeling Language™ (UML®) Specification, Object Management Group (OMG); refer to www.uml.org

The following websites provide useful reference material:

- The Cloud Computing Design Patterns community website (refer to www.cloudpatterns.org)
- The Information Technology Governance Institute: www.isaca.org/About-ISACA/IT-Governance-Institute

This website has many resources that can help with corporate assessment of both IT and governance in general.

- The Patterns-Discussion FAQ: <http://purl.org/theopengroup/pd-FAQ>

This website is maintained by Doug Lea and provides a thorough and highly readable FAQ about patterns.

Referenced Documents

- The Patterns Home Page: hillside.net/patterns

This website is hosted by The Hillside Group and provides information about patterns, links to online patterns, papers, and books dealing with patterns, and patterns-related mailing lists.

- The SOA Patterns community website (refer to www.soapatterns.org/), dedicated to the ongoing development and expansion of the SOA design pattern catalog

Referenced Documents

Appendix B: Glossary of Supplementary Definitions

This appendix contains additional definitions to supplement the definitions contained in [Chapter 4](#).

B.1 Application Software

Software entities which have a specific business purpose.

B.2 Availability

In the context of IT systems, the probability that system functional capabilities are ready for use by a user at any time, where all time is considered, including operations, repair, administration, and logistic time. Availability is further defined by system category for both routine and priority operations.

B.3 Business System

Hardware, software, policy statements, processes, activities, standards, and people which together implement a business capability.

B.4 Catalog

A structured list of architectural outputs of a similar kind, used for reference. For example, a technology standards catalog or an application portfolio.

B.5 Client

An application component which requests services from a server.

B.6 COBIT

An acronym for Control Objectives for Information and related Technology, created by the Information Systems Audit and Control Association (ISACA) and the IT Governance Institute (ITGI), which provides a set of recommended best practices for the governance/management of information systems and technology.

B.7 Configuration Management

A discipline applying technical and administrative direction and surveillance to:

- Identify and document the functional and physical characteristics of a configuration item
- Control changes to those characteristics
- Record and report changes to processing and implementation status

Also, the management of the configuration of Enterprise Architecture practice (intellectual property) assets and baselines and the control of change over of those assets.

B.8 CxO

The chief officer within a particular function of the business; e.g., Chief Executive Officer, Chief Financial Officer, Chief Information Officer, Chief Technology Officer.

B.9 Data Dictionary

A specialized type of database containing metadata; a repository of information describing the characteristics of data used to design, monitor, document, protect, and control data in information systems and databases; an application system supporting the definition and management of database metadata.

B.10 Data Element

A basic unit of information having a meaning and that may have subcategories (data items) of distinct units and values.

B.11 Database

A structured or organized collection of data entities, which is to be accessed by a computer.

B.12 Database Management System

A computer application program that accesses or manipulates the database.

B.13 End User

Person who ultimately uses the computer application or output.

B.14 Enterprise Resource Planning (ERP) System

A complete suite of integrated applications that support the major business support functions of an organization; e.g., Financial (AP/AR/GL), HR, Payroll, Stock, Order Processing and Invoicing, Purchasing, Logistics, Manufacturing, etc.

B.15 Hardware

The physical infrastructure needed to run software; e.g., servers, workstations, network equipment, etc.

B.16 Information Domain

Grouping of information (or data entities) by a set of criteria such as security classification, ownership, location, etc. In the context of security, information domains are defined as a set of users, their information objects, and a security policy.

B.17 Information System (IS)

The computer (or IT)-based portion of a business system.

B.18 Interaction

A relationship between architectural building blocks (i.e., services or components) that embodies communication or usage.

B.19 Interaction Model

An architectural view, catalog, or matrix that shows a particular type of interaction. For example, a diagram showing application integration.

B.20 Interface

Interconnection and inter-relationships between, for example, people, systems, devices, applications, or the user and an application or device.

B.21 Key Performance Indicator (KPI)

A way of quantifying the performance of the business or project.

B.22 Lifecycle

The period of time that begins when a system is conceived and ends when the system is no longer available for use.

B.23 Managing Successful Programs (MSP)

A best practice methodology for program management, developed by the UK Office of Government Commerce (OGC).

B.24 Matrix

A format for showing the relationship between two (or more) architectural elements in a grid format.

B.25 Metaview

A pattern or template of the view, from which to develop individual views. Establishes the purposes and audience for a view, the ways in which the view is documented (e.g., for visual modeling), and the ways in which it is used (e.g., for analysis).

See also [Section 4.21](#) in [Chapter 4](#).

B.26 Open System

A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered application software:

- To be ported with minimal changes across a wide range of systems
- To interoperate with other applications on local and remote systems
- To interact with users in a style that facilitates user portability

B.27 Operational Governance

The operational performance of systems against contracted performance levels, the definition of operational performance levels, and the implementation of systems that ensure effective operation of systems.

See also [Section 4.48](#) in [Chapter 4](#).

B.28 Packaged Services

Services that are acquired from the market from a Commercial Off-The-Shelf (COTS) vendor, rather than being constructed via code build.

B.29 Portability

1. The ease with which a system, component, data, or user can be transferred from one hardware or software environment to another.
2. A quality metric that can be used to measure the relative effort to transport the software for use in another environment or to convert software for use in another operating environment, hardware configuration, or software system environment.

B.30 Portfolio

A collection of programs, projects, and/or operations managed as a group to achieve strategic objectives. For example, project portfolio, application portfolio, technology portfolio, or service portfolio.

Note: Portfolio management is the act of managing portfolios.

B.31 PRINCE2

An acronym for PProjects IN Controlled Environments, which is a standard project management method.

B.32 Program

A co-ordinated set of change projects that deliver business benefit to the organization.

B.33 Project

A single change project which delivers business benefit to the organization.

B.34 Risk Management

The management of risks and issues that may threaten the success of the Enterprise Architecture practice and its ability to meet its vision, goals, and objectives, and, importantly, its service provision.

Note: Risk management is described in the TOGAF Standard — Architecture Content.

B.35 Scalability

The ability to use the same application software on many different classes of hardware/software platforms from PCs to super-computers (extends the portability concept). The capability to grow to accommodate increased work loads.

B.36 Security

Services which protect data, ensuring its confidentiality, availability, and integrity.

B.37 Server

An application component which responds to requests from a client.

B.38 Service Quality

A preset configuration of non-functional attributes that may be assigned to a service or service contract.

B.39 SMART

An acronym for Specific, Measurable, Actionable, Realistic, and Timebound, which is an approach to ensure that targets and objectives are set in a way that can be achieved and measured.

B.40 Supplier Management

The management of suppliers of products and services to the Enterprise Architecture practice in concert with larger corporate procurement activities.

B.41 System

A combination of interacting elements organized to achieve one or more stated purposes.
(Source: ISO/IEC/IEEE 15288:2015)

B.42 Time Period

The timeframe over which the potential impact is to be measured.

B.43 Use-Case

A view of organization, application, or product functionality that illustrates capabilities in context with the user of that capability.

B.44 User

1. Any person, organization, or functional unit that uses the services of an information processing system.
2. In a conceptual schema language, any person or any thing that may issue or receive commands and messages to or from the information system.

Appendix C: Abbreviations

ABB	Architecture Building Block
ACMM	Architecture Capability Maturity Model
ADM	Architecture Development Method
ANSI	American National Standards Institute
API	Application Platform Interface
ARTS	Association for Retail Technology Standards
BMM	Business Motivation Model
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BTEP	The Canadian Government Business Transformation Enablement Program
CMM	Capability Maturity Models
CMMI	Capability Maturity Model Integration
COBIT	Control Objectives for Information and related Technology
COTS	Commercial Off-The-Shelf applications
CRM	Customer Relationship Management
CRUD	Create/Read/Update/Delete
CSF	Critical Success Factor
DBA	Database Administrator
DBMS	Database Management System
DoC	US Department of Commerce
DoD	US Department of Defense
DoDAF	Department of Defense Architecture Framework
EAI	Enterprise Application Integration
EDIFACT	(United Nations) Electronic Data Interchange For Administration, Commerce, and Transport
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
FICO	Fair Isaac Corporation

FTE	Full-Time Equivalent
GOTS	Government Off-The-Shelf applications
HIPAA	Health Insurance Portability and Accountability Act
ICAM	Integrated Computer Aided Manufacturing
ICOM	Inputs, Controls, Outputs, and Mechanisms/Resources
IDEF	Integrated Computer Aided Manufacturing (ICAM) DEFinition
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
III-RM	Integrated Information Infrastructure Reference Model
IoT	Internet of Things
ISACA	Information Systems Audit and Control Association
ISACF	Information Systems Audit and Control Foundation
ISO	International Standards Organization
IT	Information Technology
ITGI	IT Governance Institute
ITIL	Information Technology Infrastructure Library
ITPMF	IT Portfolio Management Facility
J2EE	Java 2 Platform, Enterprise Edition
KPI	Key Performance Indicator
LAN	Local Area Network
MDA	Model-Driven Architecture
MSA	Microservices Architecture
MSP	Managing Successful Programs
MVA	Minimum Viable Architecture
NAF	NATO Architecture Framework
NASCIO	National Association of State Chief Information Officers
OECD	Organization for Economic Co-operation and Development
OGC	UK Office of Government Commerce
OLA	Operational-Level Agreement
OMB	Office of Management and Budget
OMG	Object Management Group
ORB	Object Request Broker
OSI	Open Systems Interconnection

Abbreviations

OSOA	Open Service-Oriented Architecture
PDF	Portable Document Format
PMBOK	Project Management Body of Knowledge
PRINCE	PRojects in Controlled Environments
QoS	Quality of Service
RAS	Remote Access Services
RFC	Request For Change
RFI	Request for Information
RFP	Request for Proposal
RM	Reference Model
SBB	Solution Building Block
SCA	Service Component Architecture
SDO	Service Data Objects
SEI	Software Engineering Institute
SGML	Standard Generalized Markup Language
SLA	Service-Level Agreement
SMART	Specific, Measurable, Actionable, Realistic, and Timebound
SOA	Service-Oriented Architecture
SPEM	Software Processing Engineering Metamodel
SysML	Systems Modeling Language
TAFIM	Technical Architecture Framework for Information Management
TRM	Technical Reference Model
UML	Unified Modeling Language
WAN	Wide Area Network
XML	Extensible Markup Language

Index

ABB.....	42
abbreviations.....	69
abstraction	23, 41
abstraction, conceptual.....	24
abstraction, contextual	24
abstraction, logical	24
abstraction, physical	24
actor	41
ADM.....	25, 43
application.....	42
Application Architecture	16, 41
Application Platform	42
Application Service	42
Application Software	61
architectural style	42
architecture	42
definition	15
Architecture Building Block	42
Architecture Capability	28, 33
Architecture Continuum	27, 43
Architecture Development Method.....	43
architecture domain	43
Architecture Forum	1
Architecture Framework	43
Architecture Governance	43
Architecture Landscape	28, 43
architecture level	44
Architecture Metamodel	28
architecture model	44
architecture partition	44
Architecture Principle	44
Architecture Principles	24-25
Architecture Repository	27
Architecture Requirements Repository	29
architecture styles	36
architecture view	37, 44
architecture viewpoint	37, 44
Architecture Vision	45
artifact	45
availability	61
baseline	45
Boundaryless Information Flow	45
building block	45
Business Architecture	15, 46

business capability	46
business function	46
business governance	46
business model	46
business service	46
business system	61
capability	46
Capability Architecture	47
capability increment	47
catalog	61
client	61
COBIT	61
communications management	34, 47
concern	47
conditions of use	6
configuration management	35, 62
Content Framework	29
core concepts	15
course of action	47
CxO	62
Data Architecture	15, 47
data dictionary	62
data element	62
database	62
database management system	62
deliverable	48
digital architecture	48
downloads	7
EAI	26
end user	62
enterprise	2, 48
enterprise agility	38
Enterprise Architecture	2
Enterprise Architecture service	48
Enterprise Architecture services	18
Enterprise Continuum	26, 48
Enterprise Metamodel	29, 31
Enterprise Principles	24
Enterprise Resource Planning system	63
environment management	35
ERP	63
financial management	34
Foundation Architecture	48
framework	48
gap	49
governance	49
Governance Repository	28
hardware	63
information	49
information domains	63

information system.....	63
information technology.....	49
interaction	63
interaction model.....	63
interface	63
interoperability.....	25, 49
IS	63
ISO/IEC/IEEE 42010: 2011	15
IT	49
iteration	39
Key Performance Indicator.....	63
KPI	63
level.....	39
lifecycle	64
logical.....	49
Managing Successful Programs	64
matrix	64
metadata	49
metamodel	49
metaview.....	64
method.....	50
model	50
model kind	50
MSP	64
MVA	20
objective.....	50
open system	64
operational governance	64
packaged services	64
partition	39
pattern.....	50
performance management.....	34
physical.....	50
portability	65
portfolio	65
PRINCE2	65
principle	50
product.....	50
program.....	65
project	65
quality management	35
Reference Library	28
reference models	51
requirement.....	51
resource management.....	34
risk management	34, 39, 65
RM	51
roadmap.....	51
role	51
SBB.....	52

scalability	65
security	66
Segment Architecture	51
server	66
service	52
service categories.....	18
service delivery model	18
service management	34
service orientation.....	52
service portfolio.....	52
service qualities	66
Service-Oriented Architecture.....	52
SMART	50, 66
SOA	52
Solution Architecture.....	52
solution building block.....	52
Solutions Continuum.....	27, 52
Solutions Landscape	29
stakeholder	53
stakeholder management	34, 47
Standards Library	28, 53
Strategic Architecture.....	53
supplier management	35, 66
system	66
TAFIM	1
Target Architecture.....	53
taxonomy of architecture views.....	53
Technology Architecture.....	16, 53
technology component.....	53
technology service	54
The Open Group.....	7
time period	66
TOGAF.....	1
TOGAF ADM.....	25
TOGAF Content Framework	29
TOGAF Library.....	10, 13
TOGAF Standard	10
Transition Architecture	54
US DoD.....	1
use-case	66
user	67
value stream	54
view	54
viewpoint.....	54
viewpoint library	54
work package.....	54

The Open Group Standard

TOGAF® Standard — Architecture Development Method

The Open Group

Copyright © 2005-2022, The Open Group

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

Any use of this publication for commercial purposes is subject to the terms of the Annual Commercial License relating to it. For further information, see www.opengroup.org/legal/licensing.

The Open Group Standard

TOGAF® Standard — Architecture Development Method

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

Chapter	1	Introduction	1
	1.1	ADM Overview.....	1
	1.1.1	The ADM, Enterprise Continuum, and Architecture Repository	1
	1.1.2	The ADM and the Foundation Architecture	2
	1.1.3	ADM and Supporting Guidelines and Techniques	2
	1.2	Architecture Development Cycle	3
	1.2.1	Key Points.....	3
	1.2.2	Basic Structure	3
	1.3	Adapting the ADM	5
	1.4	Architecture Governance	7
	1.5	Scoping the Architecture	7
	1.5.1	Breadth	9
	1.5.2	Depth	9
	1.5.3	Time Period	10
	1.5.4	Architecture Domains	11
	1.6	Architecture Alternatives.....	11
	1.6.1	Method.....	12
	1.7	Architecture Integration	13
	1.8	Summary	14
 Chapter	 2	 Preliminary Phase	 15
	2.1	Objectives	16
	2.2	Inputs.....	16
	2.2.1	Reference Materials External to the Enterprise.....	16
	2.2.2	Non-Architectural Inputs	16
	2.2.3	Architectural Inputs	17
	2.3	Steps	17
	2.3.1	Scope the Enterprise Organizations Impacted.....	18
	2.3.2	Confirm Governance and Support Frameworks	18
	2.3.3	Define and Establish Enterprise Architecture Team and Organization	18
	2.3.4	Identify and Establish Architecture Principles.....	19
	2.3.5	Tailor the TOGAF Framework and, if any, Other Selected Architecture Framework(s)	19
	2.3.6	Develop a Strategy and Implementation Plan for Tools and Techniques	20
	2.4	Outputs	21
	2.5	Approach	21
	2.5.1	Enterprise	22
	2.5.2	Organizational Context	23
	2.5.3	Requirements for Architecture Work.....	24

	2.5.4	Principles	24
	2.5.5	Management Frameworks	24
	2.5.6	Relating the Management Frameworks	26
	2.5.7	Planning for Enterprise Architecture/Business Change Maturity Evaluation	27
Chapter	3	Phase A: Architecture Vision	29
	3.1	Objectives	30
	3.2	Inputs	30
	3.2.1	Reference Materials External to the Enterprise.....	30
	3.2.2	Non-Architectural Inputs	30
	3.2.3	Architectural Inputs	30
	3.3	Steps	31
	3.3.1	Establish the Architecture Project.....	31
	3.3.2	Identify Stakeholders, Concerns, and Business Requirements	32
	3.3.3	Confirm and Elaborate Business Goals, Business Drivers, and Constraints	32
	3.3.4	Evaluate Capabilities	33
	3.3.5	Assess Readiness for Business Transformation.....	33
	3.3.6	Define Scope	34
	3.3.7	Confirm and Elaborate Architecture Principles, including Business Principles	34
	3.3.8	Develop Architecture Vision.....	34
	3.3.9	Define the Target Architecture Value Propositions and KPIs	35
	3.3.10	Identify the Business Transformation Risks and Mitigation Activities	35
	3.3.11	Develop Statement of Architecture Work; Secure Approval.....	36
	3.4	Outputs	36
	3.5	Approach	37
	3.5.1	General	37
	3.5.2	Creating the Architecture Vision	38
Chapter	4	Phase B: Business Architecture	41
	4.1	Objectives	42
	4.2	Inputs	42
	4.2.1	Reference Materials External to the Enterprise.....	42
	4.2.2	Non-Architectural Inputs	42
	4.2.3	Architectural Inputs	42
	4.3	Steps	43
	4.3.1	Select Reference Models, Viewpoints, and Tools	44
	4.3.2	Develop Baseline Business Architecture Description.....	46
	4.3.3	Develop Target Business Architecture Description.....	47
	4.3.4	Perform Gap Analysis.....	47
	4.3.5	Define Candidate Roadmap Components.....	47
	4.3.6	Resolve Impacts Across the Architecture Landscape	47
	4.3.7	Conduct Formal Stakeholder Review	48

Contents

4.3.8	Finalize the Business Architecture	48
4.3.9	Create/Update the Architecture Definition Document	48
4.4	Outputs	49
4.5	Approach	50
4.5.1	General	50
4.5.2	Developing the Baseline Description	51
4.5.3	Applying Business Capabilities.....	51
4.5.4	Applying Value Streams	52
4.5.5	Applying the Organization Map	52
4.5.6	Applying Information Maps	53
4.5.7	Applying Modeling Techniques	53
4.5.8	Architecture Repository	55
Chapter 5	Phase C: Information Systems Architectures	57
5.1	Objectives	58
5.2	Approach	58
Chapter 6	Phase C: Information Systems Architectures — Data Architecture.....	59
6.1	Objectives	59
6.2	Inputs	59
6.2.1	Reference Materials External to the Enterprise.....	59
6.2.2	Non-Architectural Inputs	59
6.2.3	Architectural Inputs.....	60
6.3	Steps	61
6.3.1	Select Reference Models, Viewpoints, and Tools	61
6.3.2	Develop Baseline Data Architecture Description	64
6.3.3	Develop Target Data Architecture Description.....	64
6.3.4	Perform Gap Analysis.....	64
6.3.5	Define Candidate Roadmap Components.....	65
6.3.6	Resolve Impacts Across the Architecture Landscape	65
6.3.7	Conduct Formal Stakeholder Review	65
6.3.8	Finalize the Data Architecture	66
6.3.9	Create/Update the Architecture Definition Document	66
6.4	Outputs	66
6.5	Approach	67
6.5.1	Data Structure	67
6.5.2	Key Considerations for Data Architecture.....	68
6.5.3	Architecture Repository	69
Chapter 7	Phase C: Information Systems Architectures — Application Architecture.....	71
7.1	Objectives	71
7.2	Inputs	71
7.2.1	Reference Materials External to the Enterprise.....	71
7.2.2	Non-Architectural Inputs	71
7.2.3	Architectural Inputs.....	71
7.3	Steps	72

7.3.1	Select Reference Models, Viewpoints, and Tools	73
7.3.2	Develop Baseline Application Architecture Description.....	76
7.3.3	Develop Target Application Architecture Description.....	76
7.3.4	Perform Gap Analysis.....	77
7.3.5	Define Candidate Roadmap Components.....	77
7.3.6	Resolve Impacts Across the Architecture Landscape	77
7.3.7	Conduct Formal Stakeholder Review	78
7.3.8	Finalize the Application Architecture	78
7.3.9	Create/Update the Architecture Definition Document	78
7.4	Outputs	78
7.5	Approach	79
7.5.1	Architecture Repository	79
Chapter 8	Phase D: Technology Architecture	81
8.1	Objectives	82
8.2	Inputs	82
8.2.1	Reference Materials External to the Enterprise.....	82
8.2.2	Non-Architectural Inputs	82
8.2.3	Architectural Inputs.....	82
8.3	Steps	83
8.3.1	Select Reference Models, Viewpoints, and Tools	84
8.3.2	Develop Baseline Technology Architecture Description.....	87
8.3.3	Develop Target Technology Architecture Description.....	88
8.3.4	Perform Gap Analysis.....	88
8.3.5	Define Candidate Roadmap Components.....	89
8.3.6	Resolve Impacts Across the Architecture Landscape	89
8.3.7	Conduct Formal Stakeholder Review	89
8.3.8	Finalize the Technology Architecture.....	89
8.3.9	Create/Update the Architecture Definition Document	90
8.4	Outputs	90
8.5	Approach	91
8.5.1	Emerging Technologies	91
8.5.2	Architecture Repository	91
Chapter 9	Phase E: Opportunities & Solutions.....	93
9.1	Objectives	94
9.2	Inputs	94
9.2.1	Reference Materials External to the Enterprise.....	94
9.2.2	Non-Architectural Inputs	94
9.2.3	Architectural Inputs.....	94
9.3	Steps	95
9.3.1	Determine/Confirm Key Corporate Change Attributes	96
9.3.2	Determine Business Constraints for Implementation	96
9.3.3	Review and Consolidate Gap Analysis Results from	

Contents

	Phases B to D.....	96
9.3.4	Review Consolidated Requirements Across Related Business Functions	97
9.3.5	Consolidate and Reconcile Interoperability Requirements	97
9.3.6	Refine and Validate Dependencies	98
9.3.7	Confirm Readiness and Risk for Business Transformation	98
9.3.8	Formulate Implementation and Migration Strategy	98
9.3.9	Identify and Group Major Work Packages	99
9.3.10	Identify Transition Architectures.....	99
9.3.11	Create the Architecture Roadmap & Implementation and Migration Plan.....	99
9.4	Outputs	100
9.5	Approach	102
Chapter 10	Phase F: Migration Planning.....	103
10.1	Objectives	104
10.2	Inputs	104
10.2.1	Reference Materials External to the Enterprise.....	104
10.2.2	Non-Architectural Inputs	104
10.2.3	Architectural Inputs.....	104
10.3	Steps	106
10.3.1	Confirm Management Framework Interactions for the Implementation and Migration Plan	106
10.3.2	Assign a Business Value to Each Work Package	107
10.3.3	Estimate Resource Requirements, Project Timings, and Availability/Delivery Vehicle	108
10.3.4	Prioritize the Migration Projects through the Conduct of a Cost/Benefit Assessment and Risk Validation	108
10.3.5	Confirm Architecture Roadmap and Update Architecture Definition Document	108
10.3.6	Complete the Implementation and Migration Plan.....	109
10.3.7	Complete the Architecture Development Cycle and Document Lessons Learned	109
10.4	Outputs	109
10.5	Approach	110
Chapter 11	Phase G: Implementation Governance.....	111
11.1	Objectives	112
11.2	Inputs	112
11.2.1	Reference Materials External to the Enterprise.....	112
11.2.2	Non-Architectural Inputs	112
11.2.3	Architectural Inputs.....	112
11.3	Steps	113
11.3.1	Confirm Scope and Priorities for Deployment with Development Management.....	114
11.3.2	Identify Deployment Resources and Skills	114
11.3.3	Guide Development of Solutions Deployment	114

11.3.4	Perform Enterprise Architecture Compliance	
	Reviews	115
11.3.5	Implement Business and IT Operations	115
11.3.6	Perform Post-Implementation Review and Close the	
	Implementation	115
11.4	Outputs	116
11.5	Approach	116
Chapter 12	Phase H: Architecture Change Management.....	119
12.1	Objectives	120
12.2	Inputs	120
12.2.1	Reference Materials External to the Enterprise.....	120
12.2.2	Non-Architectural Inputs	120
12.2.3	Architectural Inputs.....	120
12.3	Steps	122
12.3.1	Establish Value Realization Process	122
12.3.2	Deploy Monitoring Tools	122
12.3.3	Manage Risks.....	123
12.3.4	Provide Analysis for Architecture Change	
	Management.....	123
12.3.5	Develop Change Requirements to Meet Performance	
	Targets.....	123
12.3.6	Manage Governance Process	123
12.3.7	Activate the Process to Implement Change	123
12.4	Outputs	124
12.5	Approach	124
12.5.1	Drivers for Change.....	125
12.5.2	Enterprise Architecture Change Management Process	126
12.5.3	Guidelines for Maintenance versus Architecture	
	Redesign	127
Chapter 13	ADM Architecture Requirements Management.....	129
13.1	Objectives	130
13.2	Inputs	130
13.3	Steps	131
13.4	Outputs	134
13.5	Approach	135
13.5.1	General	135
13.5.2	Requirements Development	135
13.5.3	Resources	136
	Index	137
List of Figures		
1-1	Architecture Development Cycle	3
1-2	Architecture Alternatives Method.....	12
1-3	Integration of Architecture Artifacts	13
2-1	Preliminary Phase	15
2-2	Management Frameworks to Co-ordinate with the TOGAF	

Contents

	Framework	25
2-3	Interoperability and Relationships between Management Frameworks	26
3-1	Phase A: Architecture Vision	29
4-1	Phase B: Business Architecture	41
4-2	UML Business Class Diagram	54
5-1	Phase C: Information Systems Architectures	57
8-1	Phase D: Technology Architecture	81
9-1	Phase E: Opportunities & Solutions	93
10-1	Phase F: Migration Planning	103
11-1	Phase G: Implementation Governance	111
12-1	Phase H: Architecture Change Management	119
13-1	ADM Architecture Requirements Management	129

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.

Preface

This Document

This is the TOGAF Standard — Architecture Development Method.

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.

COBIT is a registered trademark of the Information Systems Audit and Control Association (ISACA) and the IT Governance Institute.

ITIL and PRINCE2 are registered trademarks of AXELOS Limited.

MDA, Model-Driven Architecture, Object Management Group, OMG, and UML are registered trademarks and BPMN, Business Process Modeling Notation, and Unified Modeling Language are trademarks of the Object Management Group.

Zachman is a registered trademark of Zachman International, Inc.

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.

Referenced Documents

Chapter 1: Introduction

This chapter introduces the Architecture Development Method (ADM) cycle, adapting the ADM, architecture scope, and architecture integration.

1.1 ADM Overview

The TOGAF ADM describes a method for developing and managing the lifecycle of an Enterprise Architecture, and forms the core of the TOGAF Standard.

It integrates elements of the TOGAF Standard, as well as other available architectural assets, to meet the business needs of an organization.

1.1.1 The ADM, Enterprise Continuum, and Architecture Repository

The Enterprise Continuum provides a framework and context to support the leverage of relevant architecture assets in executing the ADM. These assets may include Architecture Descriptions, models, and patterns taken from a variety of sources, as explained in the TOGAF Standard — Architecture Content.

The Enterprise Continuum categorizes architectural source material — both the contents of the organization's own enterprise repositories and the set of relevant, available reference models and standards in the industry.

The practical implementation of the Enterprise Continuum will typically take the form of an Architecture Repository (see the TOGAF Standard — Architecture Content) that includes reference architectures, models, and patterns that have been accepted for use within the enterprise, and actual architectural work done previously within the enterprise. The architect would seek to re-use as much as possible from the Architecture Repository that was relevant to the project in hand. (In addition to the collection of architecture source material, the repository would also contain architecture development work-in-progress.)

At relevant places throughout the ADM there are reminders to consider which, if any, architecture assets from the Architecture Repository the architect should use. In some cases — for example, in the development of a Technology Architecture — this may be the TOGAF Foundation Architecture. In other cases — for example, in the development of a Business Architecture — it may be a reference model for e-Commerce taken from the industry at large.

The criteria for including source materials in an organization's Architecture Repository will typically form part of the Enterprise Architecture Governance process. These governance processes should consider available resources both within and outside the enterprise in order to determine when general resources can be adapted for specific enterprise needs and also to determine where specific solutions can be generalized to support wider re-use.

While using the ADM, the architect is developing a snapshot of the enterprise's decisions and

their implications at particular points in time. Each iteration of the ADM will populate an organization-specific landscape with all the architecture assets identified and leveraged through the process, including the final organization-specific architecture delivered.

Architecture development is a continuous, cyclical process, and in executing the ADM repeatedly over time, the architect gradually adds more and more content to the organization's Architecture Repository. Although the primary focus of the ADM is on the development of the enterprise-specific architecture, in this wider context the ADM can also be viewed as the process of populating the enterprise's own Architecture Repository with relevant re-usable building blocks taken from the "left", more generic side of the Enterprise Continuum.

In fact, the first execution of the ADM will often be the hardest, since the architecture assets available for re-use will be relatively scarce. Even at this stage of development, however, there will be architecture assets available from external sources such as the TOGAF Standard, as well as the IT industry at large, that could be leveraged in support of the effort.

Subsequent executions will be easier as more and more architecture assets become identified, are used to populate the organization's Architecture Repository, and are thus available for future re-use.

1.1.2 The ADM and the Foundation Architecture

The ADM is also useful to populate the Foundation Architecture of an enterprise. Business requirements of an enterprise may be used to identify the necessary definitions and selections in the Foundation Architecture. This could be a set of re-usable common models, policy and governance definitions, or even as specific as overriding technology selections (e.g., if mandated by law). Population of the Foundation Architecture follows similar principles as for an Enterprise Architecture, with the difference that requirements for a whole enterprise are restricted to the overall concerns and thus less complete than for a specific enterprise.

It is important to recognize that existing models from these various sources, when integrated, may not necessarily result in a coherent Enterprise Architecture. "Integratability" of Architecture Descriptions is considered in [Section 1.7](#).

1.1.3 ADM and Supporting Guidelines and Techniques

The application of the TOGAF ADM is supported by an extended set of resources — guidelines, templates, checklists, and other detailed materials. These are included in:

- The TOGAF Standard — ADM Techniques
- TOGAF Series Guides — the Guidance part of the Standard (guidance material on how to use and adapt the TOGAF Standard for specific needs)
- White Papers and Guides published by The Open Group, classified and referenced in the TOGAF Library (see www.opengroup.org/togaf-library)

The individual guidelines and techniques are described separately so that they can be referenced from the relevant points in the ADM as necessary, rather than having the detailed text clutter the description of the ADM itself.

1.2 Architecture Development Cycle

1.2.1 Key Points

The following are the key points about the ADM:

- The ADM is iterative, over the whole process, between phases, and within phases (see the TOGAF Standard — ADM Techniques)

For each iteration of the ADM, a fresh decision must be taken as to:

- The breadth of coverage of the enterprise to be defined
- The level of detail to be defined
- The extent of the time period aimed at, including the number and extent of any intermediate time periods
- The architectural assets to be leveraged, including:
 - Assets created in previous iterations of the ADM cycle within the enterprise
 - Assets available elsewhere in the industry (other frameworks, systems models, vertical industry models, etc.)
- These decisions should be based on a practical assessment of resource and competence availability, and the value that can realistically be expected to accrue to the enterprise from the chosen scope of the architecture work
- As a generic method, the ADM is intended to be used by enterprises in a wide variety of different geographies and applied in different vertical sectors/industry types

As such, it may be, but does not necessarily have to be, tailored to specific needs. For example, it may be used in conjunction with the set of deliverables of another framework, where these have been deemed to be more appropriate for a specific organization. (For example, many US Federal agencies have developed individual frameworks that define the deliverables specific to their particular departmental needs.)

These issues are considered in detail in [Section 1.3](#).

1.2.2 Basic Structure

The basic structure of the ADM is shown in [Figure 1-1](#).

Throughout the ADM cycle, there needs to be frequent validation of results against the original expectations, both those for the whole ADM cycle, and those for the particular phase of the process.

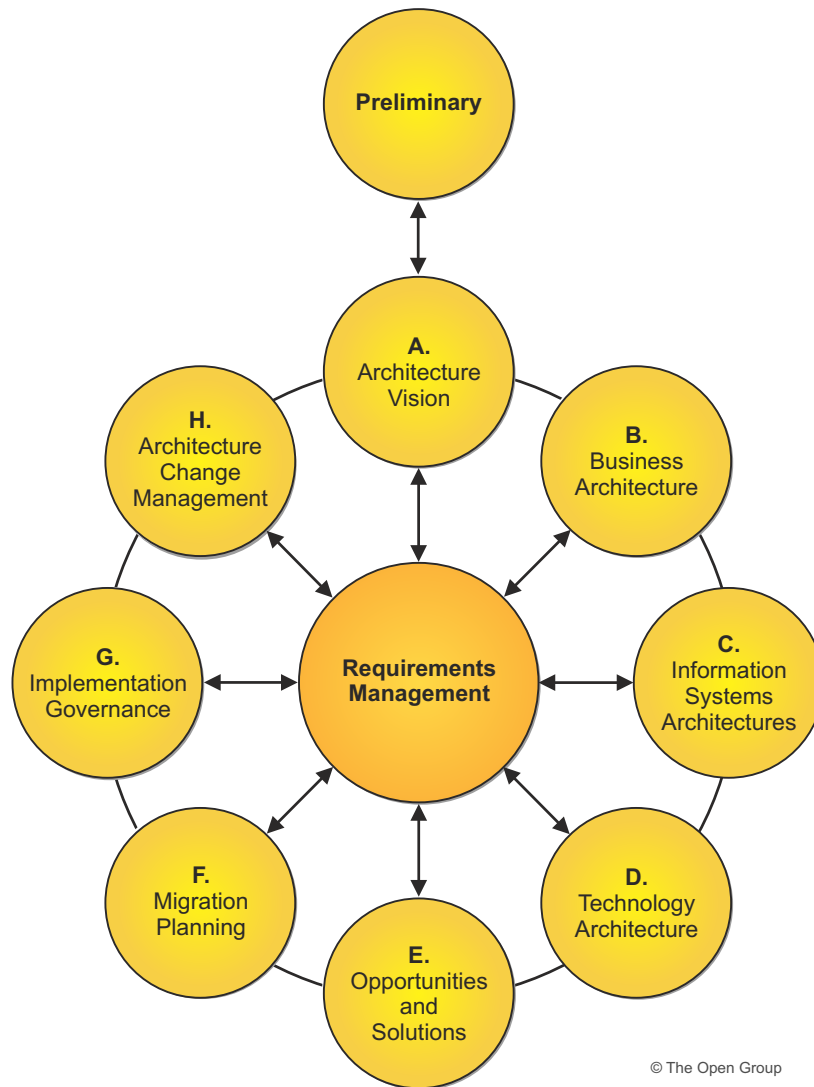


Figure 1-1 Architecture Development Cycle

The phases of the ADM cycle are further divided into steps, which are defined in the detailed description of each phase.

The Requirements Management phase is a continuous phase which ensures that any changes to requirements are handled through appropriate governance processes and reflected in all other phases. An enterprise may choose to record all new requirements, including those which are in scope of the current Statement of Architecture Work through a single Requirements Repository.

The phases of the cycle are described in detail in the following chapters.

Note that output is generated throughout the process, and that the output from an early phase may be modified in a later phase. In the ADM, the status of outputs at each stage is defined.

The lifecycle of outputs must be managed through a version numbering policy, adapted by the architect to meet the requirements of the organization and to work with the architecture tools and repositories employed by the organization.

In the ADM, documents which are under development and have not undergone any formal review and approval process are designated "draft". Documents which have been reviewed and approved are designated "approved" in accordance with the organization's governance practices. Approved does not necessarily mean finalized. Documents may evolve during subsequent phases but may only be changed through an appropriate change control and governance process. This is used in particular within the ADM to illustrate the evolution of Baseline and Target Architecture Definitions.

1.3 Adapting the ADM

The ADM is a generic method for architecture development, which is designed to deal with most system and organizational requirements. However, it will often be necessary to modify or extend the ADM to suit specific needs. One of the tasks before applying the ADM is to review its components for applicability, and then tailor them as appropriate to the circumstances of the individual enterprise. This activity may well produce an "enterprise-specific" ADM.

One reason for wanting to adapt the ADM, which it is important to stress, is that the order of the phases in the ADM is to some extent dependent on the maturity of the architecture discipline within the enterprise. For example, if the business case for doing architecture at all is not well recognized, then creating an Architecture Vision is almost always essential; and a detailed Business Architecture often needs to come next, in order to underpin the Architecture Vision, detail the business case for remaining architecture work, and secure the active participation of key stakeholders in that work. In other cases a slightly different order may be preferred; for example, a detailed inventory of the baseline environment may be done before undertaking the Business Architecture.

The order of phases may also be defined by the Architecture Principles and business principles of an enterprise. For example, the business principles may dictate that the enterprise be prepared to adjust its business processes to meet the needs of a packaged solution, so that it can be implemented quickly to enable a fast response to market changes. In such a case, the Business Architecture (or at least the completion of it) may well follow completion of the Information Systems Architecture or the Technology Architecture.

Another reason for wanting to adapt the ADM is if the TOGAF framework is to be integrated with another enterprise framework (as explained in the TOGAF Standard — Introduction and Core Concepts). For example, an enterprise may wish to use the TOGAF framework and its generic ADM in conjunction with the Zachman[®] Framework, or another Enterprise Architecture framework that has a defined set of deliverables specific to a particular vertical sector: Government, Defense, e-Business, Telecommunications, etc. The ADM has been specifically designed with this potential integration in mind.

Other possible reasons for wanting to adapt the ADM include:

- The ADM is one of the many corporate processes that make up the corporate governance model

It is complementary to, and supportive of, other standard program management processes, such as those for authorization, risk management, business planning and budgeting, development planning, systems development, and procurement.

- The ADM is being mandated for use by a prime or lead contractor in an outsourcing situation, and needs to be tailored to achieve a suitable compromise between the contractor's existing practices and the contracting enterprise's requirements
- The enterprise is a small-to-medium enterprise, and wishes to use a "cut-down" method more attuned to the reduced level of resources and system complexity typical of such an environment
- The enterprise is very large and complex, comprising many separate but interlinked "enterprises" within an overall collaborative business framework, and the architecture method needs to be adapted to recognize this

Different approaches to planning and integration may be used in such cases, including the following (possibly in combination):

- Top-down planning and development — designing the whole interconnected meta-enterprise as a single entity (an exercise that typically stretches the limits of practicality)
 - Development of a "generic" or "reference" architecture, typical of the enterprises within the organization, but not representing any specific enterprise, which individual enterprises are then expected to adapt in order to produce an architecture "instance" suited to the particular enterprise concerned
 - Replication — developing a specific architecture for one enterprise, implementing it as a proof-of-concept, and then taking that as a "reference architecture" to be cloned in other enterprises
- In a vendor or production environment, a generic architecture for a family of related products is often referred to as a "Product Line Architecture", and the analogous process to that outlined above is termed "(Architecture-based) Product Line Engineering". The ADM is targeted primarily at architects in IT user enterprises, but a vendor organization whose products are IT-based might well wish to adapt it as a generic method for a Product Line Architecture development.

The descriptions of the phases of the ADM contain lists of deliverables and artifacts that could be applicable to any enterprise. It is important to adapt deliverables and artifacts to reflect the specific needs of the enterprise for the required architecture.

Adapting the Content Framework and Enterprise Metamodel, which defines the organization-specific deliverables and artifacts, together with detailed descriptions of the specific deliverables and artifacts referenced in the ADM phase descriptions may be found in the TOGAF Standard — Architecture Content.

The ADM can also be adapted to support an Agile Enterprise Architecture delivery style as well as to enable enterprise agility. Detailed guidance on how to adapt the ADM may be found in the following documents:

- TOGAF® Series Guide: Enabling Enterprise Agility
- TOGAF® Series Guide: Applying the ADM Using Agile Sprints

Additional guidance on adapting the ADM process may be found in the TOGAF Standard — Applying the ADM, and also in the TOGAF® Series Guide: A Practitioners' Approach to Developing Enterprise Architecture Following the TOGAF ADM.

1.4 Architecture Governance

The ADM, whether adapted by the organization or used as documented here, is a key process to be managed in the same manner as other architecture artifacts classified through the Enterprise Continuum and held in the Architecture Repository. The Architecture Board should be satisfied that the method is being applied correctly across all phases of an architecture development iteration. Compliance with the ADM is fundamental to the governance of the architecture, to ensure that all considerations are made and all required deliverables are produced.

The management of all architectural artifacts, governance, and related processes should be supported by a controlled environment. Typically, this would be based on one or more repositories supporting versioned objects, process control, and status.

The major information areas managed by a governance repository should contain the following types of information:

- **Reference Data** (collateral from the organization's own repositories/Enterprise Continuum, including external data; e.g., COBIT®, the IT4IT Reference Architecture): used for guidance and instruction during project implementation

This includes the details of information outlined above. The reference data includes a description of the governance procedures themselves.

- **Process Status:** all information regarding the state of any governance processes will be managed

Examples of this include outstanding compliance requests, dispensation requests, and compliance assessments investigations.

- **Audit Information:** this will record all completed governance process actions and will be used to support:

- Key decisions and responsible personnel for any architecture project that has been sanctioned by the governance process
- A reference for future architectural and supporting process developments, guidance, and precedence

The governance artifacts and process are themselves part of the contents of the Architecture Repository.

Architecture Governance is addressed in detail in the TOGAF Standard — Enterprise Architecture Capability and Governance.

1.5 Scoping the Architecture

There are many reasons to constrain (or restrict) the scope of the architectural activity to be undertaken, most of which relate to limits in:

- The organizational authority of the team producing the architecture
- The objectives and stakeholder concerns to be addressed within the architecture
- The availability of people, finance, and other resources

The scope chosen for the architecture activity should ideally allow the work of all architects within the enterprise to be effectively governed and integrated. This requires a set of aligned

"architecture partitions" that ensure architects are not working on duplicate or conflicting activities. It also requires the definition of re-use and compliance relationships between architecture partitions.

The division of the enterprise and its architecture-related activity is discussed in more detail in the TOGAF Standard — Applying the ADM.

Four dimensions are typically used in order to define and limit the scope of an architecture:

- **Breadth:** what is the full extent of the enterprise, and what part of that extent will this architecting effort deal with?
 - Many enterprises are very large, effectively comprising a federation of organizational units that could validly be considered enterprises in their own right
 - The modern enterprise increasingly extends beyond its traditional boundaries, to embrace a fuzzy combination of traditional business enterprise combined with suppliers, customers, and partners
- **Depth:** to what level of detail should the architecting effort go?

How much architecture is "enough"? What is the appropriate demarcation between the architecture effort and other, related activities (system design, system engineering, system development)?
- **Time Period:** what is the time period that needs to be articulated for the Architecture Vision, and does it make sense (in terms of practicality and resources) for the same period to be covered in the detailed Architecture Description?

If not, how many Transition Architectures are to be defined, and what are their time periods?
- **Architecture Domains:** a complete Enterprise Architecture Description should contain all four architecture domains (Business, Data, Application, Technology), but the realities of resource and time constraints often mean there is not enough time, funding, or resources to build a top-down, all-inclusive Architecture Description encompassing all four architecture domains, even if the enterprise scope is chosen to be less than the full extent of the overall enterprise

Typically, the scope of an architecture is first expressed in terms of breadth, depth, and time. Once these dimensions are understood, a suitable combination of architecture domains can be selected that are appropriate to the problem being addressed. Techniques for using the ADM to develop a number of related architectures are discussed in the TOGAF Standard — Applying the ADM.

The four dimensions of architecture scope are explored in detail below. In each case, particularly in large-scale environments where architectures are necessarily developed in a federated manner, there is a danger of architects optimizing within their own scope of activity, instead of at the level of the overall enterprise. It is often necessary to sub-optimize in a particular area, in order to optimize at the enterprise level. The aim should always be to seek the highest level of commonality and focus on scalable and re-usable modules in order to maximize re-use at the enterprise level.

1.5.1 Breadth

One of the key decisions is the focus of the architecture effort, in terms of the breadth of overall enterprise activity to be covered (which specific business sectors, functions, organizations, geographical areas, etc.).

It is often necessary to have a number of different architectures existing across an enterprise, focused on particular timeframes, business functions, or business requirements.

For large complex enterprises, federated architectures — independently developed, maintained, and managed architectures that are subsequently integrated within an integration framework — are typical. 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.

The feasibility of a single enterprise-wide architecture for every business function or purpose may be rejected as too complex and unwieldy. In these circumstances it is suggested that a number of different Enterprise Architectures exist across an enterprise. These Enterprise Architectures focus on particular timeframes, business segments or functions, and specific organizational requirements. In such a case we need to create the overarching Enterprise Architecture as a "federation" of these Enterprise Architectures. An effective way of managing and exploiting these Enterprise Architectures is to adopt a publish-and-subscribe model that allows architecture to be brought under a governance framework. In such a model, architecture developers and architecture consumers in projects (the supply and demand sides of architecture work) sign up to a mutually beneficial framework of governance that ensures that:

- Architectural material is of good quality, up-to-date, fit-for-purpose, and published (reviewed and agreed to be made public)
- Usage of architecture material can be monitored, and compliance with standards, models, and principles can be exhibited, via:
 - A Compliance Assessment process that describes what the user is subscribing to, and assesses their level of compliance
 - A dispensation process that may grant dispensations from adherence to architecture standards and guidelines in specific cases (usually with a strong business imperative)

Publish and subscribe techniques are being developed as part of general IT governance and specifically for the Defense sphere.

1.5.2 Depth

Care should be taken to judge the appropriate level of detail to be captured, based on the intended use of the Enterprise Architecture and the decisions to be made based on it. It is important that a consistent and equal level of depth be completed in each architecture domain (Business, Data, Application, Technology) included in the architecture effort. If pertinent detail is omitted, the architecture may not be useful. If unnecessary detail is included, the architecture effort may exceed the time and resources available, and/or the resultant architecture may be confusing or cluttered. Developing architectures at different levels of detail within an enterprise is discussed in more detail in the TOGAF Standard — Applying the ADM.

It is also important to predict the future uses of the architecture so that, within resource limitations, the architecture can be structured to accommodate future tailoring, extension, or re-

use. The depth and detail of the Enterprise Architecture needs to be sufficient for its purpose, and no more.

Iterations of the ADM will build on the artifacts and the capabilities created during previous iterations.

There is a need to document all the models in an enterprise, to the level of detail appropriate to the need of the current ADM cycle. The key is to understand the status of the enterprise's architecture work, and what can realistically be achieved with the resources and competencies available, and then focus on identifying and delivering the value that is achievable. Stakeholder value is a key focus: too broad a scope may deter some stakeholders (no return on investment).

1.5.3 Time Period

The ADM is described in terms of a single cycle of Architecture Vision, and a set of Target Architectures (Business, Data, Application, Technology) that enable the implementation of the vision.

In such cases, a wider view may be taken, whereby an enterprise is represented by several different architecture instances (for example, strategic, segment, capability), each representing the enterprise at a particular point in time. One architecture instance will represent the current enterprise state (the "as-is", or baseline). Another architecture instance, perhaps defined only partially, will represent the ultimate target end-state (the "vision"). In-between, intermediate or "Transition Architecture" instances may be defined, each comprising its own set of Target Architecture Descriptions. An example of how this might be achieved is given in the TOGAF Standard — ADM Techniques.

By this approach, the Target Architecture work is split into two or more discrete stages:

1. First, develop Target Architecture Descriptions for the overall (large-scale) system, demonstrating a response to stakeholder objectives and concerns for a relatively distant timeframe (for example, a six-year period).
2. Then develop one or more "Transition Architecture" descriptions, as increments or plateaus, each in line with and converging on the Target Architecture Descriptions, and describing the specifics of the increment concerned.

In such an approach, the Target Architectures are evolutionary in nature, and require periodic review and update according to evolving business requirements and developments in technology, whereas the Transition Architectures are (by design) incremental in nature, and in principle should not evolve during the implementation phase of the increment, in order to avoid the "moving target" syndrome. This, of course, is only possible if the implementation schedule is under tight control and relatively short (typically less than two years).

The Target Architectures remain relatively generic, and are therefore less vulnerable to obsolescence than the Transition Architectures. They embody only the key strategic architectural decisions, which should be blessed by the stakeholders from the outset, whereas the detailed architectural decisions in the Transition Architectures are deliberately postponed as far as possible (i.e., just before implementation) in order to improve responsiveness *vis a vis* new technologies and products.

The enterprise evolves by migrating to each of these Transition Architectures in turn. As each Transition Architecture is implemented, the enterprise achieves a consistent, operational state on the way to the ultimate vision. However, this vision itself is periodically updated to reflect

changes in the business and technology environment, and in effect may never actually be achieved, as originally described. The whole process continues for as long as the enterprise exists and continues to change.

Such a breakdown of the Architecture Description into a family of related architecture products of course requires effective management of the set and their relationships.

1.5.4 Architecture Domains

A complete Enterprise Architecture should address all four architecture domains (Business, Data, Application, Technology), but the realities of resource and time constraints often mean there is not enough time, funding, or resources to build a top-down, all-inclusive Architecture Description encompassing all four architecture domains.

Architecture Descriptions will normally be built with a specific purpose in mind — a specific set of business drivers that drive the architecture development — and clarifying the specific issue(s) that the Architecture Description is intended to help explore, and the questions it is expected to help answer, is an important part of the initial phase of the ADM.

For example, if the purpose of a particular architecture effort is to define and examine technology options for achieving a particular capability, and the fundamental business processes are not open to modification, then a full Business Architecture may well not be warranted. However, because the Data, Application, and Technology Architectures build on the Business Architecture, the Business Architecture still needs to be thought through and understood.

While circumstances may sometimes dictate building an Architecture Description not containing all four architecture domains, it should be understood that such an architecture cannot, by definition, be a complete Enterprise Architecture. One of the risks is a lack of consistency and therefore an ability to integrate. Integration either needs to come later — with its own costs and risks — or the risks and trade-offs involved in not developing a complete and integrated architecture need to be articulated by the architect, and communicated to and understood by the enterprise management.

1.6 Architecture Alternatives

There is often more than one possible Target Architecture that would conform to the Architecture Vision, Architecture Principles, and Requirements. It is important to identify alternative Target Architectures and build understanding of different possibilities and identify trade-offs between the alternatives. Creating an architecture normally requires trade-offs among competing forces. Presenting different alternatives and trade-offs to stakeholders helps architects to extract hidden agendas, principles, and requirements that could impact the final Target Architecture.

1.6.1 Method

It is most common that a single alternative does not exist that will meet all stakeholders' concerns. The TOGAF Standard (see the TOGAF Standard — ADM Techniques: Architecture Alternatives and Trade-offs) provides a technique to investigate different alternatives and to discuss these with the stakeholders.

Commonly, alternatives are defined per domain. This is done to simplify the analysis of the different alternatives. Of course, the alternatives per domain can be merged into an overall analysis of the alternatives for the whole architecture. This approach is illustrated in [Figure 1-2](#).

The first part of the method uses the vision, principles, requirements, and other information to select sets of criteria fitting for different alternatives.

The second part of the method defines alternatives based on the criteria and builds an understanding of each.

The third part of the method will either select one of the alternatives, or else combine features from more than one, to create the proposed alternative. Perform the following activities in just enough detail to support that decision. The method can be used for any phase at any level of an architecture.

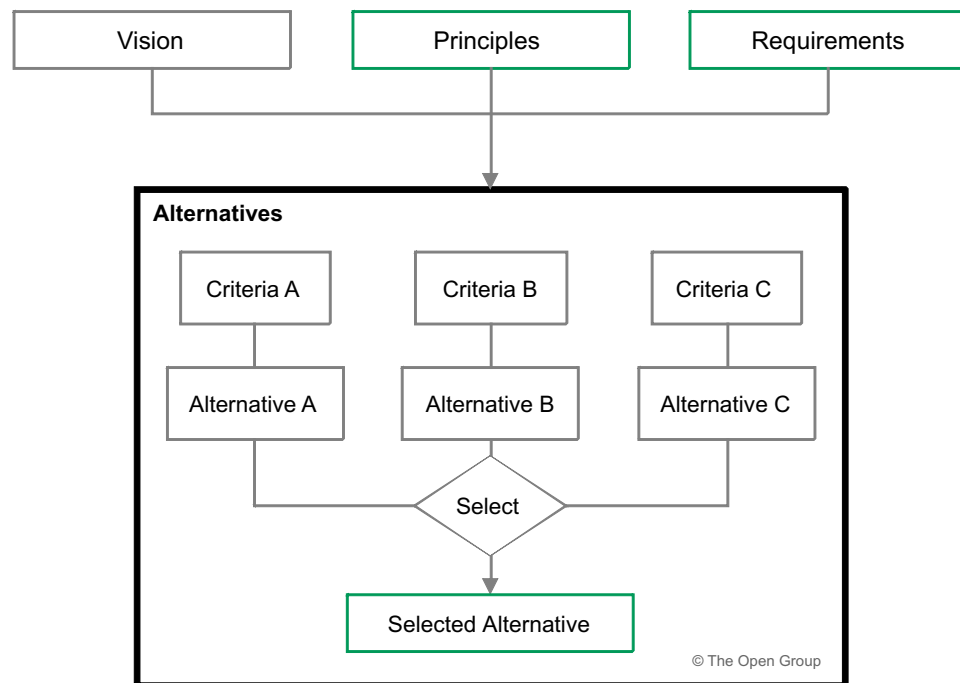


Figure 1-2 Architecture Alternatives Method

1.7 Architecture Integration

Architectures that are created to address a subset of issues within an enterprise require a consistent frame of reference so that they can be considered as a group as well as point deliverables. The dimensions that are used to define the scope boundary of a single architecture (e.g., level of detail, architecture domain, etc.) are typically the same dimensions that must be addressed when considering the integration of many architectures. [Figure 1-3](#) illustrates how different types of architecture need to co-exist.

At the present time, the state of the art is such that architecture integration can be accomplished only at the lower end of the integrability spectrum. Key factors to consider are the granularity and level of detail in each artifact, and the maturity of standards for the interchange of architectural descriptions.

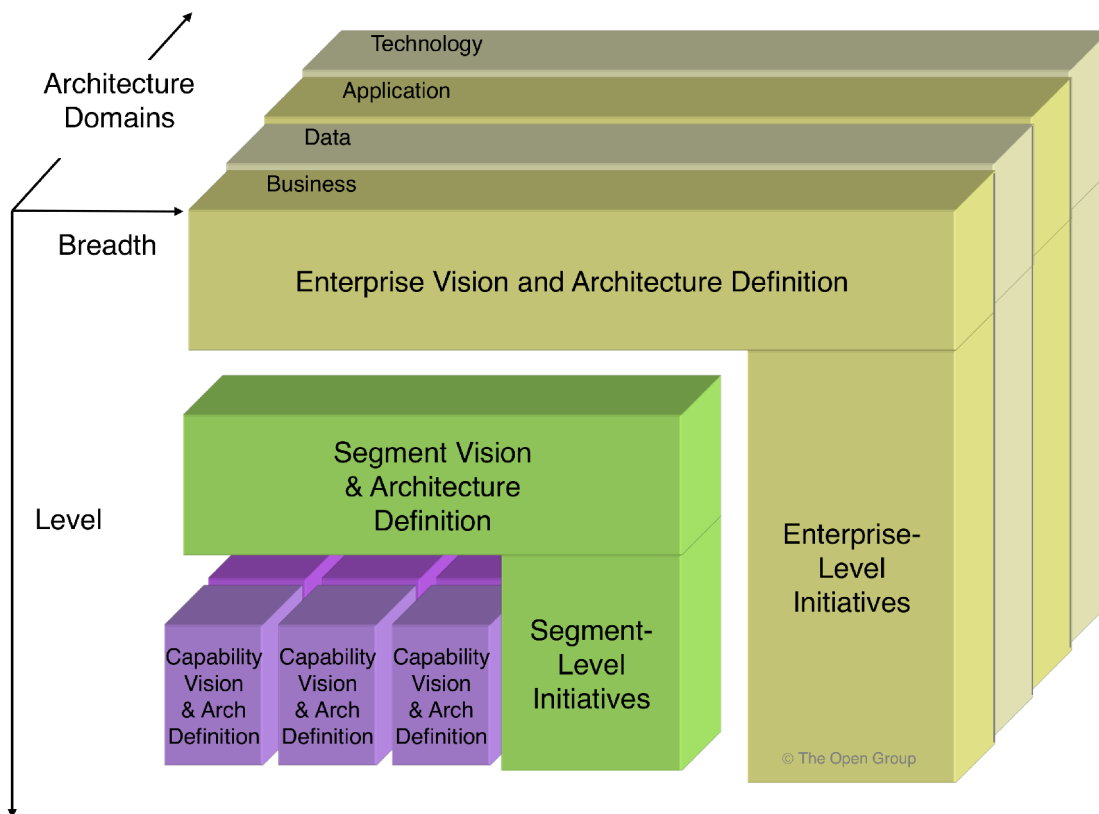


Figure 1-3 Integration of Architecture Artifacts

As organizations address common themes (such as Service-Oriented Architecture (SOA), and integrated information infrastructure) and universal data models and standard data structures emerge, integration toward the high end of the spectrum will be facilitated. However, there will always be the need for effective standards governance to reduce the need for manual co-ordination and conflict resolution.

1.8 Summary

The TOGAF ADM defines a recommended sequence for the various phases and steps involved in developing an architecture, but it cannot recommend a scope — this has to be determined by the organization itself, bearing in mind that the recommended sequence of development in the ADM process is an iterative one, with the depth and breadth of scope and deliverables increasing with each iteration. Each iteration will add resources to the organization's Architecture Repository.

While a complete framework is useful (indeed, essential) to have in mind as the ultimate long-term goal, in practice there is a key decision to be made as to the scope of a specific Enterprise Architecture effort. This being the case, it is vital to understand the basis on which scoping decisions are being made, and to set clear expectations for the goal of the effort.

The main guideline is to focus on what creates value to the enterprise, and to select horizontal and vertical scope, and time periods, accordingly. Whether or not this is the first time around, understand that this exercise will be repeated, and that future iterations will build on what is being created in the current effort, adding greater width and depth.

Chapter 2: Preliminary Phase

This chapter describes the preparation and initiation activities required to meet the business directive for a new Enterprise Architecture, including the definition of an Organization-Specific Architecture framework and the definition of principles.

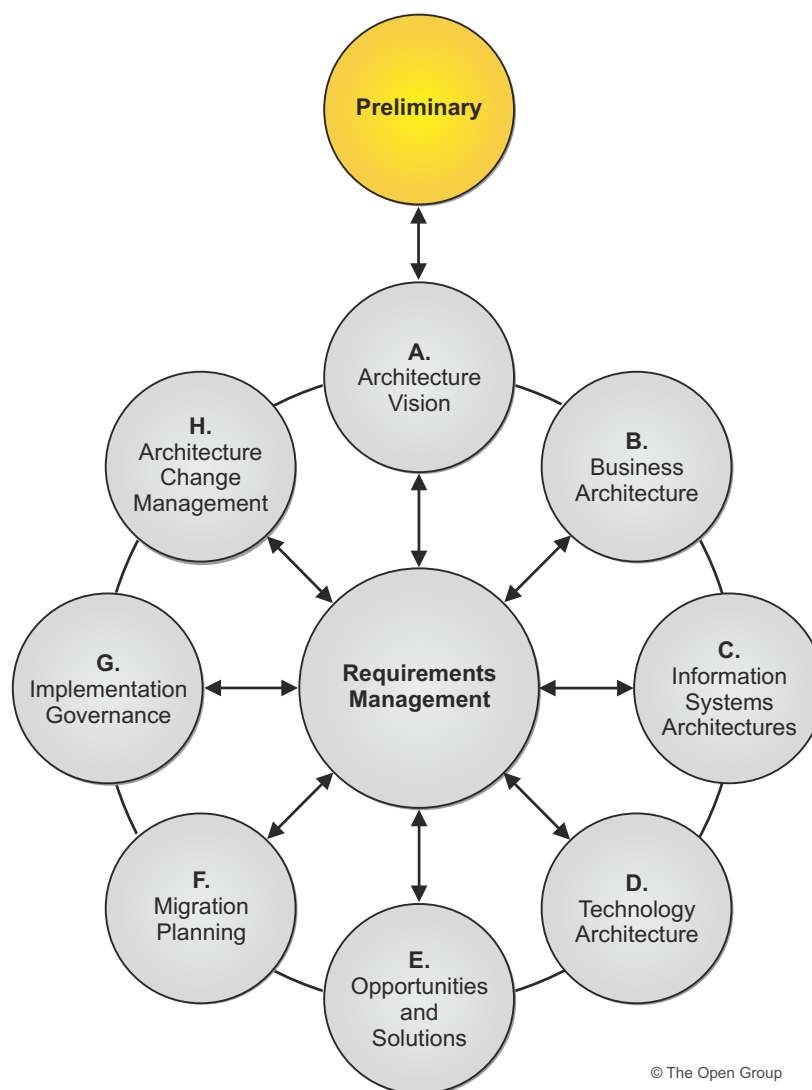


Figure 2-1 Preliminary Phase

2.1 Objectives

The objectives of the Preliminary Phase are to:

1. Determine the Architecture Capability desired by the organization:
 - Review the organizational context for conducting Enterprise Architecture
 - Identify and scope the elements of the enterprise organizations affected by the Architecture Capability
 - Identify the established frameworks, methods, and processes that intersect with the Architecture Capability
 - Establish Capability Maturity target
2. Establish the Architecture Capability:
 - Define and establish the Organizational Model for Enterprise Architecture
 - Define and establish the detailed process and resources for Architecture Governance
 - Select and implement tools that support the Architecture Capability
 - Define the Architecture Principles

2.2 Inputs

This section defines the inputs to the Preliminary Phase.

2.2.1 Reference Materials External to the Enterprise

- The TOGAF Library
- Other architecture framework(s), if required

2.2.2 Non-Architectural Inputs

- Board strategies and board business plans, business strategy, IT strategy, business principles, business goals, and business drivers, when pre-existing
- Major frameworks operating in the business; e.g., project/portfolio management
- Governance and legal frameworks, including Architecture Governance strategy, when pre-existing
- Architecture capability
- Partnership and contract agreements

2.2.3 Architectural Inputs

Pre-existing models for operating an Enterprise Architecture Capability can be used as a baseline for the Preliminary Phase. Inputs would include:

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Budget requirements
 - Governance and support strategy
- Existing Architecture Framework, if any, including:
 - Architecture method
 - Architecture content
 - Configured and deployed tools
 - Architecture Principles
 - Architecture Repository

2.3 Steps

The TOGAF ADM is a generic method, intended to be used by a wide variety of different enterprises, and in conjunction with a wide variety of other architecture frameworks, if required. The Preliminary Phase therefore involves doing any necessary work to initiate and adapt the ADM to define an organization-specific framework. The issues involved with adapting the ADM to a specific organizational context are discussed in detail in [Section 1.3](#).

The level of detail addressed in the Preliminary Phase will depend on the scope and goals of the overall architecture effort.

The order of the steps in the Preliminary Phase as well as the time at which they are formally started and completed should be adapted to the situation at hand in accordance with the established Architecture Governance.

The steps within the Preliminary Phase are as follows:

- Scope the enterprise organizations impacted (see [Section 2.3.1](#))
- Confirm governance and support frameworks (see [Section 2.3.2](#))
- Define and establish Enterprise Architecture team and organization (see [Section 2.3.3](#))
- Identify and establish Architecture Principles (see [Section 2.3.4](#))
- Tailor the TOGAF framework and, if any, other selected architecture frameworks (see [Section 2.3.5](#))

- Develop a strategy and implementation plan for tools and techniques (see [Section 2.3.6](#))

2.3.1 Scope the Enterprise Organizations Impacted

- Identify core enterprise (units) — those who are most affected and achieve most value from the work
- Identify soft enterprise (units) — those who will see change to their capability and work with core units but are otherwise not directly affected
- Identify extended enterprise (units) — those units outside the scoped enterprise who will be affected in their own Enterprise Architecture
- Identify communities involved (enterprises) — those stakeholders who will be affected and who are in groups of communities
- Identify governance involved, including legal frameworks and geographies (enterprises)

2.3.2 Confirm Governance and Support Frameworks

The architecture framework will form the keystone to the flavor (centralized or federated, light or heavy, etc.) of Architecture Governance organization and guidelines that need to be developed. Part of the major output of this phase is a framework for Architecture Governance. We need to understand how architectural material (standards, guidelines, models, compliance reports, etc.) is brought under governance; i.e., what type of governance repository characteristics are going to be required, what relationships and status recording are necessary to ascertain which governance process (dispensation, compliance, take-on, retirement, etc.) has ownership of an architectural artifact.

It is likely that the existing governance and support models of an organization will need to change to support the newly adopted architecture framework.

To manage the organizational change required to adopt the new architectural framework, the current enterprise governance and support models will need to be assessed to understand their overall shape and content. Additionally, the sponsors and stakeholders for architecture will need to be consulted on potential impacts that could occur.

Upon completion of this step, the architecture touch-points and likely impacts should be understood and agreed by relevant stakeholders.

2.3.3 Define and Establish Enterprise Architecture Team and Organization

- Determine existing enterprise and business capability
- Conduct an Enterprise Architecture/business change maturity assessment, if required
- Identify gaps in existing work areas
- Allocate key roles and responsibilities for Enterprise Architecture Capability management and governance

- Define requests for change to existing business programs and projects:
 - Inform existing Enterprise Architecture and IT architecture work of stakeholder requirements
 - Request assessment of impact on their plans and work
 - Identify common areas of interest
 - Identify any critical differences and conflicts of interest
 - Produce requests for change to stakeholder activities
- Determine constraints on Enterprise Architecture work
- Review and agree with sponsors and board
- Assess budget requirements

2.3.4 Identify and Establish Architecture Principles

Architecture Principles (see the TOGAF Standard — ADM Techniques) are based on business principles and are critical in setting the foundation for Architecture Governance. Once the organizational context is understood, define a set of Architecture Principles that is appropriate to the enterprise.

2.3.5 Tailor the TOGAF Framework and, if any, Other Selected Architecture Framework(s)

In this step, determine what tailoring of the TOGAF framework is required. Consider the need for:

- **Terminology Tailoring:** architecture practitioners should use terminology that is generally understood across the enterprise

Tailoring should produce an agreed terminology set for description of architectural content. Consideration should be given to the creation of an Enterprise Glossary, to be updated throughout the architecture process.
- **Process Tailoring:** the TOGAF ADM provides a generic process for carrying out architecture

Process tailoring provides the opportunity to remove tasks that are already carried out elsewhere in the organization, add organization-specific tasks (such as specific checkpoints), and to align the ADM processes to external process frameworks and touch-points. Key touch-points to be addressed would include:

 - Links to (project and service) portfolio management processes
 - Links to project lifecycle
 - Links to operations handover processes
 - Links to operational management processes (including configuration management, change management, and service management)

— Links to procurement processes

- **Content Tailoring:** using the TOGAF Architecture Content Framework and Enterprise Continuum as a basis, tailoring of content structure and classification approach allows adoption of third-party content frameworks and also allows for customization of the framework to support organization-specific requirements

2.3.6 Develop a Strategy and Implementation Plan for Tools and Techniques

There are many tools and techniques which may be used to develop Enterprise Architecture across many domains. The development of a tools strategy is recommended that reflects the understanding and level of formality required by the enterprise's stakeholders. Architecture content will be highly dependent on the scale, sophistication, and culture of both the stakeholders and the Architecture Capability within the organization. A tools strategy which recognizes the stakeholders' articulation requirements will enable more effective and rapid decision-making by stakeholders and their ownership of artifacts.

The strategy should encompass management techniques, decision management, workshop techniques, business modeling, detailed infrastructure modeling, office products, languages, and repository management as well as more formal architecture tools. For example, the Balanced Scorecard technique is a best practice performance measurement tool used by business schools and many organizations that can be used successfully in architecture projects.

The implementation of the tools strategy may be based on common desktop and office tools or may be based on a customized deployment of specialist management and architecture tools. Change management of the artifact deliverables is a major consideration and a degree of management control and governance of artifacts needs to be considered. Access to decisions needs to be managed carefully as many of the artifacts may contain sensitive information. Therefore the tools implementation, access, and security of the content needs to reflect the sensitivity requirements.

2.3.6.1 Issues in Tools Standardization

In the current state of the tools market, many enterprises developing Enterprise Architectures struggle with the issue of standardizing on tools, whether they seek a single "one size fits all" tool or a multi-tool suite for modeling architectures and generating the different architecture views required.

There are ostensible advantages associated with selecting a single tool. Organizations following such a policy can hope to realize benefits such as reduced training, shared licenses, quantity discounts, maintenance, and easier data interchange. However, there are also reasons for refusing to identify a single mandated tool, including reasons of principle (endorsing a single architecture tool would not encourage competitive commercial innovation or the development of advanced tool capability); and the fact that a single tool would not accommodate a variety of architecture development "maturity levels" and specific needs across an enterprise.

Successful Enterprise Architecture teams are often those that harmonize their architecture tools with their architecture maturity level, team/organizational capabilities, and objectives or focus. If different organizations within an enterprise are at different architecture maturity levels and have different objectives or focus (e.g., Enterprise *versus* Business *versus* Technology Architecture), it becomes very difficult for one tool to satisfy all organizations' needs.

2.4 Outputs

The outputs of the Preliminary Phase may include, but are not restricted to:

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Budget requirements
 - Governance and support strategy
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)
 - Architecture Principles (see the TOGAF Standard — Architecture Content)
 - Configured and deployed tools
- Initial Architecture Repository (see the TOGAF Standard — Architecture Content), populated with framework content
- Restatement of, or reference to, business principles, business goals, and business drivers (see the TOGAF Standard — Architecture Content)
- Request for Architecture Work (optional) (see the TOGAF Standard — Architecture Content)
- Architecture Governance Framework (see the TOGAF Standard — Enterprise Architecture Capability and Governance)
- The Architecture of the Enterprise Architecture Capability (see the TOGAF Standard — Enterprise Architecture Capability and Governance)

The outputs may include some or all of the following:

- Catalogs:
 - Principles catalog

2.5 Approach

This Preliminary Phase is about defining "where, what, why, who, and how we do architecture" in the enterprise concerned. The main aspects are as follows:

- Defining the enterprise
- Identifying key drivers and elements in the organizational context

- Defining the requirements for architecture work
- Defining the Architecture Principles that will inform any architecture work
- Defining the framework to be used
- Defining the relationships between management frameworks
- Evaluating the Enterprise Architecture maturity

The Enterprise Architecture provides a strategic, top-down view of an organization to enable executives, planners, architects, and engineers to coherently co-ordinate, integrate, and conduct their activities. The Enterprise Architecture framework provides the strategic context within which this team can operate.

Therefore, developing the Enterprise Architecture is not a solitary activity and the Enterprise Architects need to recognize the interoperability between their frameworks and the rest of the business.

Strategic, interim, and tactical business objectives and aspirations need to be met. Similarly, the Enterprise Architecture needs to reflect this requirement and allow for operation of architecture discipline at different levels within the organization.

Depending on the scale of the enterprise and the level of budgetary commitment to Enterprise Architecture discipline, a number of approaches may be adopted to sub-divide or partition architecture teams, processes, and deliverables. Approaches for architecture partitioning are discussed in the TOGAF Standard — Architecture Content. The Preliminary Phase should be used to determine the desired approach to partitioning and to establish the groundwork for the selected approach to be put into practice.

The Preliminary Phase may be revisited, from the Architecture Vision phase (see the TOGAF Standard — Applying the ADM), in order to ensure that the organization's Architecture Capability is suitable to address a specific architecture problem.

The Preliminary Phase is about creating an architecture for the Enterprise Architecture Capability in an organization. As such, the same best practices employed in developing any architecture should be used here. Refer to the TOGAF Standard — Enterprise Architecture Capability and Governance for an explanation about the application of the TOGAF ADM to establish the Enterprise Architecture Capability.

Further guidance on how to construct an Enterprise Architecture Capability may be found in the following document:

- TOGAF® Series Guide: The TOGAF Leader's Guide to Establishing and Evolving an EA Capability

2.5.1 Enterprise

One of the main challenges of Enterprise Architecture is that of enterprise scope.

The scope of the enterprise, and whether it is federated, will determine those stakeholders who will derive most benefit from the Enterprise Architecture Capability. It is imperative that a sponsor is appointed at this stage to ensure that the resultant activity has resources to proceed and the clear support of the business management. The enterprise may encompass many organizations and the duties of the sponsor are to ensure that all stakeholders are included in defining, establishing, and using the Architecture Capability.

2.5.2 Organizational Context

In order to make effective and informed decisions about the framework for architecture to be used within a particular enterprise, it is necessary to understand the context surrounding the architecture framework. Specific areas to consider would include:

- The commercial models for Enterprise Architecture and budgetary plans for Enterprise Architecture activity; where no such plans exist, the Preliminary Phase should be used to develop a budget plan
- The stakeholders for architecture in the enterprise; their key issues and concerns
- The intentions and culture of the organization, as captured within board business directives, business imperatives, business strategies, business principles, business goals, and business drivers
- Current processes that support execution of change and operation of the enterprise, including the structure of the process and also the level of rigor and formality applied within the organization

Areas for focus should include:

- Current methods for architecture description
- Current project management frameworks and methods
- Current systems management frameworks and methods
- Current project portfolio management processes and methods
- Current application portfolio management processes and methods
- Current technology portfolio management processes and methods
- Current information portfolio management processes and methods
- Current systems design and development frameworks and methods
- The Baseline Architecture landscape, including the state of the enterprise and also how the landscape is currently represented in documentation form
- The skills and capabilities of the enterprise and specific organizations that will be adopting the framework

Review of the organizational context should provide valuable requirements on how to tailor the architecture framework in terms of:

- Level of formality and rigor to be applied
- Level of sophistication and expenditure required
- Touch-points with other organizations, processes, roles, and responsibilities
- Focus of content coverage

2.5.3 Requirements for Architecture Work

The business imperatives behind the Enterprise Architecture work drive the requirements and performance metrics for the architecture work. They should be sufficiently clear so that this phase may scope the business outcomes and resource requirements, and define the outline enterprise business information requirements and associated strategies of the Enterprise Architecture work to be done. For example, these may include:

- Business requirements
- Cultural aspirations
- Organization intents
- Strategic intent
- Forecast financial requirements

Significant elements of these need to be articulated so that the sponsor can identify all the key decision-makers and stakeholders involved in defining and establishing an Architecture Capability.

2.5.4 Principles

The Preliminary Phase defines the Architecture Principles that will form part of the constraints on any architecture work undertaken in the enterprise. The issues involved in this are explained in the TOGAF Standard — ADM Techniques.

The definition of Architecture Principles is fundamental to the development of an Enterprise Architecture. Architecture work is informed by business principles as well as Architecture Principles. The Architecture Principles themselves are also normally based in part on business principles. Defining business principles normally lies outside the scope of the architecture function. However, depending on how such principles are defined and promulgated within the enterprise, it may be possible for the set of Architecture Principles to also restate, or cross-refer to a set of business principles, business goals, and strategic business drivers defined elsewhere within the enterprise. Within an architecture project, the architect will normally need to ensure that the definitions of these business principles, goals, and strategic drivers are current, and to clarify any areas of ambiguity.

The issue of Architecture Governance is closely linked to that of Architecture Principles. The body responsible for governance will also normally be responsible for approving the Architecture Principles, and for resolving architecture issues. The issues involved in governance are explained in the TOGAF Standard — Enterprise Architecture Capability and Governance.

2.5.5 Management Frameworks

The TOGAF ADM is a generic method, intended to be used by enterprises in a wide variety of industry types and geographies. It is also designed for use with a wide variety of other Enterprise Architecture frameworks, if required (although it can be used perfectly well in its own right, without adaptation).

The TOGAF framework has to co-exist with and enhance the operational capabilities of other management frameworks that are present within any organization either formally or informally. In addition to these frameworks, most organizations have a method for the development of

solutions, most of which have an IT component. The significance of systems is that they bring together the various domains (also known as People, Processes, and Material/Technology) to deliver a business capability.

The main frameworks suggested to be co-ordinated with the TOGAF framework are:

- **Business Capability Management** that determines what business capabilities are required to deliver business value including the definition of return on investment and the requisite control/performance measures
- **Project/Portfolio Management Methods** that determine how a company manages its change initiatives
- **Operations Management Methods** that describe how a company runs its day-to-day operations, including IT
- **Solution Development Methods** that formalize the way that business systems are delivered in accordance with the structures developed in the IT architecture

As illustrated in [Figure 2-2](#), these frameworks are not discrete and there are significant overlaps between them and the Business Capability Management. The latter includes the delivery of performance measured business value.

The overall significance is that the Enterprise Architect applying the TOGAF framework cannot narrowly focus on the IT implementation, but must be aware of the impact that the architecture has on the entire enterprise.

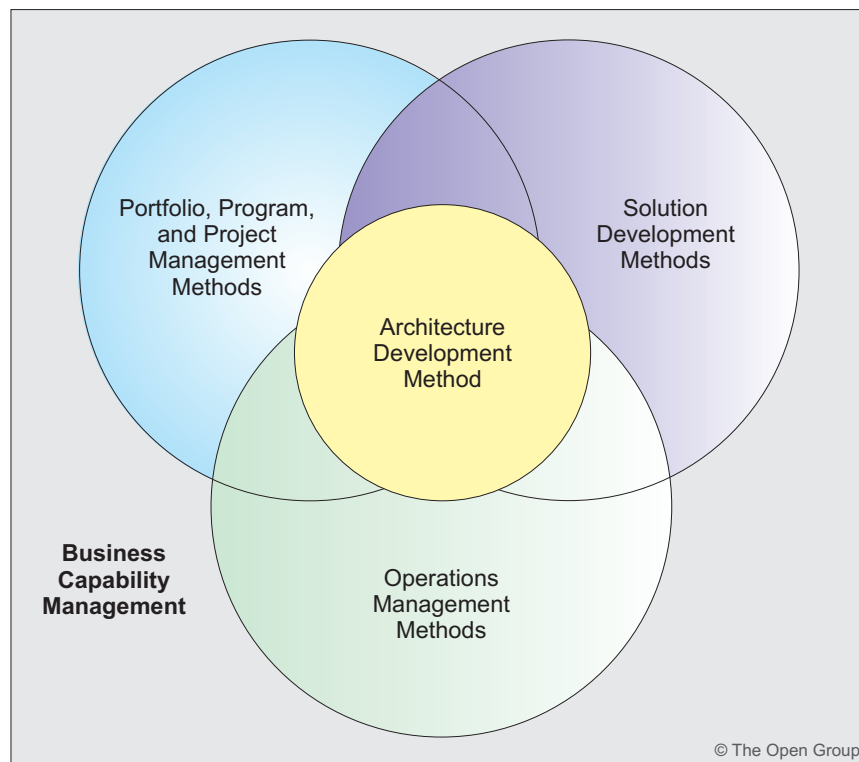


Figure 2-2 Management Frameworks to Co-ordinate with the TOGAF Framework

The Preliminary Phase therefore involves doing any necessary work to adapt the ADM to define an organization-specific framework, using either the TOGAF deliverables or the deliverables of another framework. The issues involved in this are discussed in [Section 1.3](#).

2.5.6 Relating the Management Frameworks

[Figure 2-3](#) illustrates a more detailed set of dependencies between the various frameworks and business planning activity that incorporates the enterprise's strategic plan and direction. The Enterprise Architecture can be used to provide a structure for all of the corporate initiatives, the Portfolio Management Framework can be used to deliver the components of the architecture, and the Operations Management Framework supports incorporation of these new components within the corporate infrastructure.

The business planners are present throughout the process and are in a position to support and enforce the architecture by retaining approval for resources at the various stages of planning and development.

The solution development methodology is used within the Portfolio Management Framework to plan, create, and deliver the architectural components specified in the project and portfolio charters. These deliverables include, but are not exclusively, IT; for example, a new building, a new set of skills, production equipment, hiring, marketing, and so on. Enterprise Architecture potentially provides the context for all enterprise activities.

The management frameworks are required to complement each other and work in close harmony for the good of the enterprise.

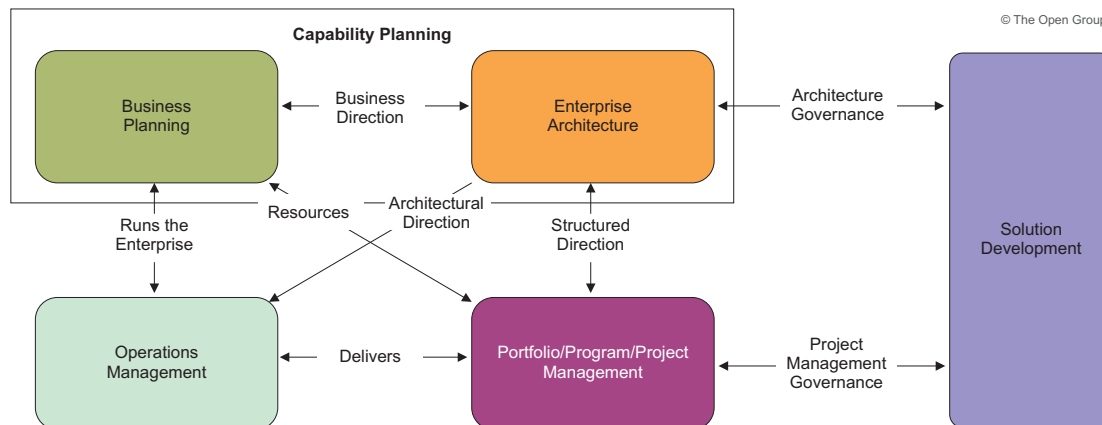


Figure 2-3 Interoperability and Relationships between Management Frameworks

Business planning at the strategy level provides the initial direction to Enterprise Architecture. Updates at the annual planning level provide a finer level of ongoing guidance. Capability-based planning is one of many popular techniques for business planning.

Enterprise Architecture structures the business planning into an integrated framework that regards the enterprise as a system or system of systems. This integrated approach will validate the business plan and can provide valuable feedback to the corporate planners. In some organizations, the Enterprise Architects have been moved to or work very closely with the

strategic direction groups. The TOGAF approach delivers a framework for Enterprise Architecture.

Project/portfolio management is the delivery framework that receives the structured, detailed direction that enables them to plan and build what is required, knowing that each assigned deliverable will be in context (i.e., the piece of the puzzle that they deliver will fit into the corporate puzzle that is the Enterprise Architecture). Often this framework is based upon the Project Management Institute or UK Office of Government Commerce (PRINCE2®) project management methodologies. Project architectures and detailed out-of-context design are often based upon systems design methodologies.

Guidance on how to use the TOGAF ADM along with a Project Management framework may be found in the TOGAF® Series Guide: Architecture Project Management.

Operations management receives the deliverables and then integrates and sustains them within the corporate infrastructure. Often the IT service management services are based upon the IT4IT Reference Architecture, ITIL®, and ISO/IEC 20000:2011.

2.5.7 Planning for Enterprise Architecture/Business Change Maturity Evaluation

Capability Maturity Models are useful ways of assessing the ability of an enterprise to exercise different capabilities.

Capability Maturity Models typically identify selected factors that are required to exercise a capability. An organization's ability to execute specific factors provides a measure of maturity and can be used to recommend a series of sequential steps to improve a capability. It is an assessment that gives executives an insight into pragmatically improving a capability.

A good Enterprise Architecture maturity model covers the characteristics necessary to develop and consume Enterprise Architecture. Organizations can determine their own factors and derive the appropriate maturity models, but it is recommended to take an existing model and customize it as required.

Several good models exist, including National Association of State CIOs (NASCIO), and the US Department of Commerce Architecture Capability Maturity Model. Other examples include the US Federal Enterprise Architecture Maturity Model. Even though the models are originally from government, they are equally applicable to industry.

Chapter 3: Phase A: Architecture Vision

This chapter describes the initial phase of the ADM. It includes information about defining the scope, identifying the stakeholders, creating the Architecture Vision, and obtaining approvals.

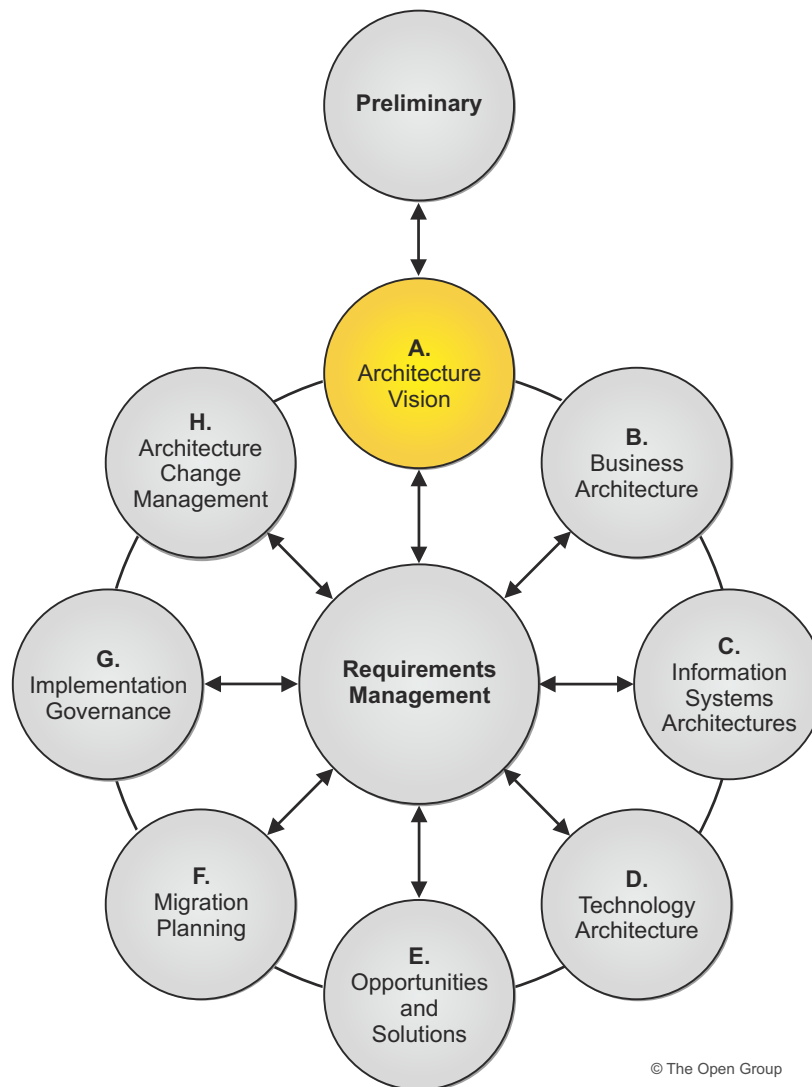


Figure 3-1 Phase A: Architecture Vision

3.1 Objectives

The objectives of Phase A are to:

- Develop a high-level aspirational vision of the capabilities and business value to be delivered as a result of the proposed Enterprise Architecture
- Obtain approval for a Statement of Architecture Work that defines a program of works to develop and deploy the architecture outlined in the Architecture Vision

3.2 Inputs

This section defines the inputs to Phase A.

3.2.1 Reference Materials External to the Enterprise

- Architecture reference materials (see the TOGAF Standard — Architecture Content)

3.2.2 Non-Architectural Inputs

- Request for Architecture Work (see the TOGAF Standard — Architecture Content)
- Business principles, business goals, and business drivers (see the TOGAF Standard — Architecture Content)

3.2.3 Architectural Inputs

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Re-use requirements
 - Budget requirements
 - Requests for change
 - Governance and support strategy
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)

- Architecture Principles (see the TOGAF Standard — Architecture Content), including business principles, when pre-existing
- Configured and deployed tools
- Populated Architecture Repository (see the TOGAF Standard — Architecture Content) — existing architectural documentation (framework description, architectural descriptions, baseline descriptions, Architecture Building Blocks (ABBs), etc.)

3.3 Steps

The level of detail addressed in Phase A will depend on the scope and goals of the Request for Architecture Work, or the subset of scope and goals associated with this iteration of architecture development.

The order of the steps in Phase A as well as the time at which they are formally started and completed should be adapted to the situation at hand in accordance with the established Architecture Governance.

The steps in Phase A are as follows:

- Establish the architecture project (see [Section 3.3.1](#))
- Identify stakeholders, concerns, and business requirements (see [Section 3.3.2](#))
- Confirm and elaborate business goals, business drivers, and constraints (see [Section 3.3.3](#))
- Evaluate capabilities (see [Section 3.3.4](#))
- Assess readiness for business transformation (see [Section 3.3.5](#))
- Define scope (see [Section 3.3.6](#))
- Confirm and elaborate Architecture Principles, including business principles (see [Section 3.3.7](#))
- Develop Architecture Vision (see [Section 3.3.8](#))
- Define the Target Architecture value propositions and KPIs (see [Section 3.3.9](#))
- Identify the business transformation risks and mitigation activities (see [Section 3.3.10](#))
- Develop Statement of Architecture Work; secure approval (see [Section 3.3.11](#))

3.3.1 Establish the Architecture Project

Enterprise Architecture is a business capability; each cycle of the ADM should normally be handled as a project using the project management framework of the enterprise. In some cases, architecture projects will be stand-alone. In other cases, architectural activities will be a subset of the activities within a larger project. In either case, architecture activity should be planned and managed using accepted practices for the enterprise.

Conduct the necessary procedures to secure recognition of the project, the endorsement of corporate management, and the support and commitment of the necessary line management. Include references to other management frameworks in use within the enterprise, explaining how this project relates to those frameworks.

3.3.2 Identify Stakeholders, Concerns, and Business Requirements

Identify the key stakeholders and their concerns/objectives, and define the key business requirements to be addressed in the architecture engagement. Stakeholder engagement at this stage is intended to accomplish three objectives:

- To identify candidate vision components and requirements to be tested as the Architecture Vision is developed
- To identify candidate scope boundaries for the engagement to limit the extent of architectural investigation required
- To identify stakeholder concerns, issues, and cultural factors that will shape how the architecture is presented and communicated

The major product resulting from this step is a stakeholder map for the engagement, showing which stakeholders are involved with the engagement, their level of involvement, and their key concerns (see the TOGAF Standard — ADM Techniques). The stakeholder map is used to support various outputs of the Architecture Vision phase, and to identify:

- The concerns and viewpoints that are relevant to this project; this is captured in the Architecture Vision (see the TOGAF Standard — Architecture Content)
- The stakeholders that are involved with the project and as a result form the starting point for a Communications Plan (see the TOGAF Standard — Architecture Content)
- The key roles and responsibilities within the project, which should be included within the Statement of Architecture Work (see the TOGAF Standard — Architecture Content)

Another key task will be to consider which architecture views and viewpoints need to be developed to satisfy the various stakeholder requirements. As described in the TOGAF Standard — ADM Techniques, understanding at this stage which stakeholders and which views need to be developed is important in setting the scope of the engagement.

During the Architecture Vision phase, new requirements generated for future architecture work within the scope of the selected requirements need to be documented within the Architecture Requirements Specification, and new requirements which are beyond the scope of the selected requirements must be input to the Requirements Repository for management through the Requirements Management process.

3.3.3 Confirm and Elaborate Business Goals, Business Drivers, and Constraints

Identify the business goals and strategic drivers of the organization.

If these have already been defined elsewhere within the enterprise, ensure that the existing definitions are current, and clarify any areas of ambiguity. Otherwise, go back to the originators of the Statement of Architecture Work and work with them to define these essential items and secure their endorsement by corporate management.

Define the constraints that must be dealt with, including enterprise-wide constraints and project-specific constraints (time, schedule, resources, etc.). The enterprise-wide constraints may be informed by the business and Architecture Principles developed in the Preliminary Phase or clarified as part of Phase A.

3.3.4 Evaluate Capabilities

It is valuable to understand a collection of capabilities within the enterprise. One part refers to the capability of the enterprise to develop and consume the architecture. The second part refers to the baseline and target capability level of the enterprise. Gaps identified in the Architecture Capability require iteration between Architecture Vision and Preliminary Phase to ensure that the Architecture Capability is suitable to address the scope of the architecture project (see the TOGAF Standard — Applying the ADM).

A key step following from evaluation of business models, or artifacts that clarify priorities of a business strategy, is to identify the required business capabilities the enterprise must possess to act on the strategic priorities.

The detailed assessment of business capability gaps belongs in Phase B as a core aspect of the Business Architecture, where the architect can help the enterprise understand gaps throughout the business, of many types, that need to be addressed in later phases of the architecture.

In the Architecture Vision phase, however, the architect should consider the capability of the enterprise to develop the Enterprise Architecture itself, as required in the specific initiative or project underway. Gaps in the ability to progress through the ADM, whether deriving from skill shortages, information required, process weakness, or systems and tools, are a serious consideration in the vision of whether the architecture effort should continue. The architect can find guidance in [Section 3.5](#) to gather existing business capability frameworks for the enterprise in this early assessment.

Gaps, or limitations, identified in the enterprise's capability to execute on change will inform the architect on the description of the Target Architecture and on the Implementation and Migration Plan (see the TOGAF Standard — Architecture Content) created in Phase E and Phase F. This step seeks to understand the capabilities and desires of the enterprise at an appropriate level of abstraction (see the TOGAF Standard — Applying the ADM). Consideration of the gap between the baseline and target capability of the enterprise is critical. Showing the baseline and target capabilities within the context of the overall enterprise can be supported by creating Value Chain diagrams that show the linkage of related capabilities. The results of the assessment are documented in a Capability Assessment (see the TOGAF Standard — Architecture Content).

3.3.5 Assess Readiness for Business Transformation

A Business Transformation Readiness Assessment can be used to evaluate and quantify the organization's readiness to undergo a change. This assessment is based upon the determination and analysis/rating of a series of readiness factors, as described in the TOGAF Standard — ADM Techniques.

The results of the readiness assessment should be added to the Capability Assessment (see the TOGAF Standard — Architecture Content). These results are then used to shape the scope of the architecture, to identify activities required within the architecture project, and to identify risk areas to be addressed.

3.3.6 Define Scope

Define what is inside and what is outside the scope of the Baseline Architecture and Target Architecture efforts, understanding that the baseline and target need not be described at the same level of detail. In many cases, the baseline is described at a higher level of abstraction, so more time is available to specify the target in sufficient detail. The issues involved in this are discussed in [Section 1.5](#). In particular, define:

- The breadth of coverage of the enterprise
- The level of detail required
- The partitioning characteristics of the architecture (see the TOGAF Standard — Applying the ADM for more details)
- The specific architecture domains to be covered (Business, Data, Application, Technology)
- The extent of the time period aimed at, plus the number and extent of any intermediate time period
- The architectural assets to be leveraged, or considered for use, from the organization's Architecture Repository:
 - Assets created in previous iterations of the ADM cycle within the enterprise
 - Assets available elsewhere in the industry (other frameworks, systems models, vertical industry models, etc.)

3.3.7 Confirm and Elaborate Architecture Principles, including Business Principles

Review the principles under which the architecture is to be developed. Architecture Principles are normally based on the principles developed as part of the Preliminary Phase. They are explained, and an example set given, in the TOGAF Standard — ADM Techniques. Ensure that the existing definitions are current, and clarify any areas of ambiguity. Otherwise, go back to the body responsible for Architecture Governance and work with them to define these essential items for the first time and secure their endorsement by corporate management.

3.3.8 Develop Architecture Vision

An understanding of the required artifacts will enable the stakeholders to start to scope out their decision-making which will guide subsequent phases. These decisions need to be reflected in the stakeholder map.

Policy development and strategic decisions need to be captured in this phase to enable the subsequent work to be quantified; for example, rationalization decisions and metrics, revenue generation, and targets which meet the business strategy. There are also other areas which need to be addressed; for example, Digital Transformation and IT strategy where decisions on the Architecture Vision will provide leadership and direction for the organization in subsequent phases.

For the Architecture Vision it is recommended that first an overall architecture be decided upon showing how all of the various architecture domain deliverables will fit together (based upon the selected course of action).

Based on the stakeholder concerns, business capability requirements, scope, constraints, and

principles, create a high-level view of the Baseline and Target Architectures. The Architecture Vision typically covers the breadth of scope identified for the project, at a high level. Informal techniques are often employed. A common practice is to draw a simple solution concept diagram that illustrates concisely the major components of the solution and how the solution will result in benefit for the enterprise.

Business scenarios are an appropriate and useful technique to discover and document business requirements, and to articulate an Architecture Vision that responds to those requirements. Business scenarios may also be used at more detailed levels of the architecture work (e.g., in Phase B) and are described in the TOGAF® Series Guide: Business Scenarios.

This step generates the first, very high-level definitions of the baseline and target environments, from a business, information systems, and technology perspective, as described in [Section 3.4](#).

These initial versions of the architecture should be stored in the Architecture Repository, organized according to the standards and guidelines established in the architecture framework.

3.3.9 Define the Target Architecture Value Propositions and KPIs

- Develop the business case for the architectures and changes required
- Produce the value proposition for each of the stakeholder groupings
- Assess and define the procurement requirements
- Review and agree the value propositions with the sponsors and stakeholders concerned
- Define the performance metrics and measures to be built into the Enterprise Architecture to meet the business needs
- Assess the business risk (see the TOGAF Standard — ADM Techniques)

The outputs from this activity should be incorporated within the Statement of Architecture Work to allow performance to be tracked accordingly.

3.3.10 Identify the Business Transformation Risks and Mitigation Activities

Identify the risks associated with the Architecture Vision and assess the initial level of risk (e.g., catastrophic, critical, marginal, or negligible) and the potential frequency associated with it. Assign a mitigation strategy for each risk. A risk management framework is described in the TOGAF Standard — ADM Techniques.

There are two levels of risk that should be considered, namely:

- **Initial Level of Risk:** risk categorization prior to determining and implementing mitigating actions
- **Residual Level of Risk:** risk categorization after implementation of mitigating actions (if any)

Risk mitigation activities should be considered for inclusion within the Statement of Architecture Work.

3.3.11 Develop Statement of Architecture Work; Secure Approval

Assess the work products that are required to be produced (and by when) against the set of business performance requirements. This will involve ensuring that:

- Performance metrics are built into the work products
- Specific performance-related work products are available

Then, activities will include:

- Identify new work products that will need to be changed
- Provide direction on which existing work products, including building blocks, will need to be changed and ensure that all activities and dependencies on these are co-ordinated
- Identify the impact of change on other work products and dependence on their activities
- Based on the purpose, focus, scope, and constraints, determine which architecture domains should be developed, to what level of detail, and which architecture views should be built
- Assess the resource requirements and availability to perform the work in the timescale required; this will include adhering to the organization's planning methods and work products to produce the plans for performing a cycle of the ADM
- Estimate the resources needed, develop a roadmap and schedule for the proposed development, and document all these in the Statement of Architecture Work
- Define the performance metrics to be met during this cycle of the ADM by the Enterprise Architecture team
- Develop the specific Enterprise Architecture Communications Plan and show where, how, and when the Enterprise Architects will communicate with the stakeholders, including affinity groupings and communities, about the progress of the Enterprise Architecture developments
- Review and agree the plans with the sponsors, and secure formal approval of the Statement of Architecture Work under the appropriate governance procedures
- Gain sponsor's sign-off to proceed

3.4 Outputs

The outputs of Phase A may include, but are not restricted to:

- Approved Statement of Architecture Work (see the TOGAF Standard — Architecture Content), including in particular:
 - Architecture project description and scope
 - Overview of Architecture Vision
 - Architecture project plan and schedule
- Refined statements of business principles, business goals, and business drivers (see the TOGAF Standard — Architecture Content)

- Architecture Principles (see the TOGAF Standard — ADM Techniques)
- Capability Assessment (see the TOGAF Standard — Architecture Content)
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content) (for the engagement), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)
 - Configured and deployed tools
- Architecture Vision (see the TOGAF Standard — Architecture Content), including:
 - Problem description
 - Objective of the Statement of Architecture Work
 - Summary views
 - Business scenario (optional)
 - Refined key high-level stakeholder requirements
- Draft Architecture Definition Document, which may include Baseline and/or Target Architectures of any architecture domain
- Communications Plan (see the TOGAF Standard — Architecture Content)
- Additional content populating the Architecture Repository (see the TOGAF Standard — Architecture Content)

Note: Multiple business scenarios may be used to generate a single Architecture Vision.

The TOGAF Standard — Architecture Content contains a detailed description of architectural artifacts which might be produced in this phase, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

3.5 Approach

3.5.1 General

Phase A starts with receipt of a Request for Architecture Work from the sponsoring organization to the architecture organization.

The issues involved in ensuring proper recognition and endorsement from corporate management, and the support and commitment of line management, are discussed in the TOGAF Standard — Enterprise Architecture Capability and Governance.

Phase A also defines what is in and what is outside the scope of the architecture effort and the constraints that must be dealt with. Scoping decisions need to be made on the basis of a practical assessment of resource and competence availability, and the value that can realistically be expected to accrue to the enterprise from the chosen scope of architecture work. The issues involved in this are discussed in [Section 1.5](#). Scoping issues addressed in the Architecture Vision phase will be restricted to the specific objectives for this ADM cycle and will be constrained within the overall scope definition for architecture activity as established within the Preliminary Phase and embodied within the architecture framework.

In situations where the architecture framework in place is not appropriate to achieve the desired Architecture Vision, revisit the Preliminary Phase and extend the overall architecture framework for the enterprise.

The constraints will normally be informed by the business principles and Architecture Principles, developed as part of the Preliminary Phase (see [Chapter 2](#)).

Normally, the business principles, business goals, and strategic drivers of the organization are already defined elsewhere in the enterprise. If so, the activity in Phase A is involved with ensuring that existing definitions are current, and clarifying any areas of ambiguity. Otherwise, it involves defining these essential items for the first time.

Similarly, the Architecture Principles that form part of the constraints on architecture work will normally have been defined in the Preliminary Phase (see [Chapter 2](#)). The activity in Phase A is concerned with ensuring that the existing principle definitions are current, and clarifying any areas of ambiguity. Otherwise, it entails defining the Architecture Principles for the first time, as explained in the TOGAF Standard — ADM Techniques.

3.5.2 Creating the Architecture Vision

The Architecture Vision provides the sponsor with a key tool to sell the benefits of the proposed capability to stakeholders and decision-makers within the enterprise. Architecture Vision describes how the new capability will meet the business goals and strategic objectives and address the stakeholder concerns when implemented.

Integral to the Architecture Vision is an understanding of emerging technologies and their potential impact on industries and enterprises, without which many business opportunities may be missed.

Clarifying and agreeing the purpose of the architecture effort is one of the key parts of this activity, and the purpose needs to be clearly reflected in the vision that is created. Architecture projects are often undertaken with a specific purpose in mind — a specific set of business drivers that represent the return on investment for the stakeholders in the architecture development. Clarifying that purpose, and demonstrating how it will be achieved by the proposed architecture development, is the whole point of the Architecture Vision.

Normally, key elements of the Architecture Vision — such as the enterprise mission, vision, strategy, and goals — have been documented as part of some wider business strategy or enterprise planning activity that has its own lifecycle within the enterprise. In such cases, the activity in Phase A is concerned with verifying and understanding the documented business strategy and goals, and possibly bridging between the enterprise strategy and goals on the one hand, and the strategy and goals implicit within the current architecture reality.

Business models are key strategy artifacts that can provide such a perspective, by showing how the organization intends to deliver value to its customers and stakeholders. [Section 3.3.4](#) introduces the application of business models as a step in developing the Architecture Vision.

In other cases, little or no Business Architecture work may have been done to date. In such cases, there will be a need for the architecture team to research, verify, and gain buy-in to the key business objectives and processes that the architecture is to support. This may be done as a free-standing exercise, either preceding architecture development, or as part of the ADM initiation phase (Preliminary Phase).

This exercise should examine and search for existing materials on fundamental Business Architecture concepts such as:

- **Business Capabilities**, which represent a particular ability or capacity that a business may possess or exchange to achieve a specific purpose or outcome

In this phase, the architect should determine whether a framework exists in the organization to represent business capabilities. If one does not exist, the architect should consider whether developing a framework is within the scope of the project. For an introduction to business capabilities, see TOGAF® Series Guide: Business Capabilities.

- **Value Streams**, which represent an end-to-end collection of value-adding activities that create an overall result for a customer, stakeholder, or end user

For an introduction to value streams, see the TOGAF® Series Guide: Value Streams.

- **Organization Maps**, which depict the relationships between the primary entities that make up the enterprise, its partners, and stakeholders

As traditional organizational charts often lack the necessary detail to reflect the full scope of the enterprise's activities, the architect can help to identify and understand the complex web of relationships between business entities as well as where business capabilities are used and connection to value stream stages. These are refined and extended in subsequent phases. For an introduction to organization maps, see the TOGAF® Series Guide: Organization Mapping.

In addition, the Architecture Vision explores other domains which are appropriate for the Enterprise Architecture in hand. These domains may include elements of the basic domains, yet serve an additional purpose for the stakeholders. Example domains may include:

- Information
- Security
- Digital
- Network Management
- Knowledge
- Industry-specific
- Services
- Partnership
- Cybersecurity

These domains may be free-standing or linked with other domains to provide enterprise-wide views of the organization vision and structure.

The Architecture Vision phase includes the conduct of a business assessment (using, for example, business scenarios) where critical factors are documented and various courses of action are assessed. High-level advantages and disadvantages, including risks and opportunities, are documented and the best course of action selected to serve as the basis for the Architecture Vision.

The Architecture Vision provides a first-cut, high-level description of the Baseline and Target Architectures, covering the Business, Data, Application, and Technology domains. These outline descriptions are developed in subsequent phases.

Once an Architecture Vision is defined and documented in the Statement of Architecture Work, it is critical to use it to build a consensus, as described in the TOGAF Standard — Enterprise Architecture Capability and Governance. Without this consensus it is very unlikely that the final architecture will be accepted by the organization as a whole. The consensus is represented by the sponsoring organization signing the Statement of Architecture Work.

Chapter 4: Phase B: Business Architecture

This chapter describes the development of a Business Architecture to support an agreed Architecture Vision.

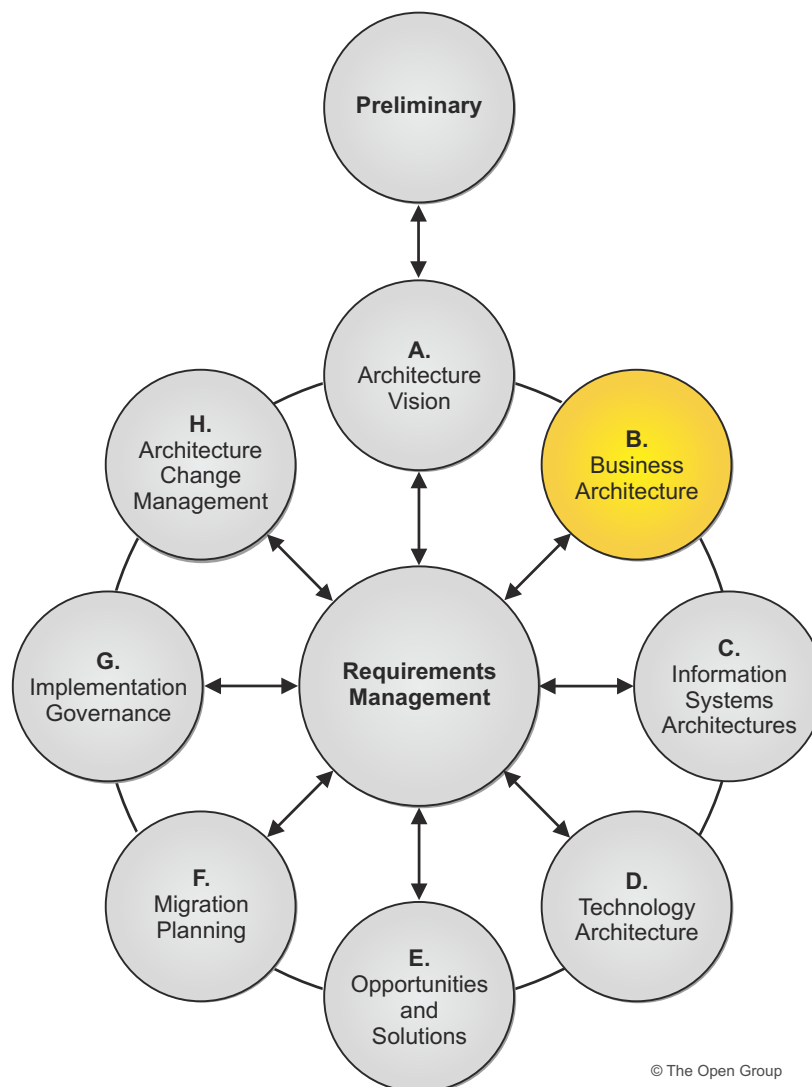


Figure 4-1 Phase B: Business Architecture

4.1 Objectives

The objectives of Phase B are to:

- 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, in a way that addresses the Statement of Architecture Work and stakeholder concerns
- Identify candidate Architecture Roadmap components based upon gaps between the Baseline and Target Business Architectures

4.2 Inputs

This section defines the inputs to Phase B.

4.2.1 Reference Materials External to the Enterprise

- Architecture reference materials (see the TOGAF Standard — Architecture Content)

4.2.2 Non-Architectural Inputs

- Request for Architecture Work (see the TOGAF Standard — Architecture Content)
- Business principles, business goals, and business drivers (see the TOGAF Standard — Architecture Content)
- Capability Assessment (see the TOGAF Standard — Architecture Content)
- Communications Plan (see the TOGAF Standard — Architecture Content)

4.2.3 Architectural Inputs

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Budget requirements
 - Governance and support strategy
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method

- Tailored architecture content (deliverables and artifacts)
- Configured and deployed tools
- Approved Statement of Architecture Work (see the TOGAF Standard — Architecture Content)
- Architecture Principles (see the TOGAF Standard — Architecture Content), including business principles, when pre-existing
- Enterprise Continuum (see the TOGAF Standard — Architecture Content)
- Architecture Repository (see the TOGAF Standard — Architecture Content), including:
 - Re-usable building blocks
 - Publicly available reference models
 - Organization-specific reference models
 - Organization standards
- Architecture Vision (see the TOGAF Standard — Architecture Content), including:
 - Problem description
 - Objective of the Statement of Architecture Work
 - Summary views
 - Business Scenario (optional)
 - Refined key high-level stakeholder requirements
- Draft Architecture Definition Document, which may include Baseline and/or Target Architectures of any architecture domain

4.3 Steps

The level of detail addressed in Phase B will depend on the scope and goals of the overall architecture effort.

New models characterizing the needs of the business will need to be defined in detail during Phase B. Existing business artifacts to be carried over and supported in the target environment may already have been adequately defined in previous architectural work; but, if not, they too will need to be defined in Phase B.

The order of the steps in Phase B as well as the time at which they are formally started and completed should be adapted to the situation at hand, in accordance with the established Architecture Governance. In particular, determine whether in this situation it is appropriate to conduct Baseline or Target Architecture development first, as described in the TOGAF Standard — Applying the ADM.

All activities that have been initiated in these steps should be closed during the Finalize the Business Architecture step (see [Section 4.3.8](#)). The documentation generated from these steps must be formally published in the Create/Update the Architecture Definition Document step (see [Section 4.3.9](#)).

The steps in Phase B are as follows:

- Select reference models, viewpoints, and tools (see [Section 4.3.1](#))
- Develop Baseline Business Architecture Description (see [Section 4.3.2](#))
- Develop Target Business Architecture Description (see [Section 4.3.3](#))
- Perform gap analysis (see [Section 4.3.4](#))
- Define candidate roadmap components (see [Section 4.3.5](#))
- Resolve impacts across the Architecture Landscape (see [Section 4.3.6](#))
- Conduct formal stakeholder review (see [Section 4.3.7](#))
- Finalize the Business Architecture (see [Section 4.3.8](#))
- Create/Update the Architecture Definition Document (see [Section 4.3.9](#))

4.3.1 Select Reference Models, Viewpoints, and Tools

Select relevant Business Architecture resources (reference models, patterns, etc.) from the Architecture Repository, on the basis of the business drivers and stakeholder concerns.

Select relevant Business Architecture viewpoints (e.g., operations, management, financial); i.e., those that will enable the architect to demonstrate how the stakeholder concerns are being addressed in the Business Architecture.

Identify appropriate tools and techniques to be used for capture, modeling, and analysis, in association with the selected viewpoints. Depending on the degree of sophistication warranted, these may comprise simple documents or spreadsheets, or more sophisticated modeling tools and techniques, such as activity models, business process models, use-case models, etc.

4.3.1.1 Determine Overall Modeling Process

Business modeling and strategy assessments are effective techniques for framing the target state of an organization's Business Architecture (see [Section 3.3.4](#)). The output from that activity is then used to articulate the business capabilities, organizational structure, and value streams required to close gaps between the current and target state. As addressed in [Section 3.5](#), the frameworks for these maps may already exist and should be leveraged, now using them to characterize gaps and better map business value to achieve the Target Business Architecture.

For each viewpoint, select the models needed to support the specific view required, using the selected tool or method.

Ensure that all stakeholder concerns are covered. If they are not, create new models to address concerns not covered, or augment existing models (see [Section 4.5.7](#)). Business scenarios are a useful technique to discover and document business requirements, and may be used iteratively, at different levels of detail in the hierarchical decomposition of the Business Architecture. Business scenarios are described in the TOGAF® Series Guide: Business Scenarios.

The techniques described in [Section 4.5](#) can be utilized to progressively decompose a business:

- **Business Capability Mapping:** identifies, categorizes, and decomposes the business capabilities required for the business to have the ability to deliver value to one or more stakeholders
- **Information Mapping:** collecting the information concepts and their relationships that matter most to the business
- **Organization Mapping:** a representation of the organizational structure of the business (including third-party domains), depicting business units, the decomposition of those units into lower-level functions, and organizational relationships (unit-to-unit and mapping to business capabilities, locations, and other attributes)
- **Process Modeling:** the activity of articulating business processes of an enterprise, to enable analysis and improvement
- **Structured Analysis:** identifies the key business capabilities within the scope of the architecture, and maps those capabilities onto business functions and organizational units within the business
- **Use-case Analysis:** a technique used to identify the requirements of a system or task to be completed, from a user's perspective
- **Value Stream Mapping:** the end-to-end breakdown of activities that an organization performs to create the value being exchanged with stakeholders

Value stream maps illustrate how an organization delivers value and are in the context of a specific set of stakeholders, and leverage business capabilities in order to create stakeholder value and align to other aspects of the Target Business Architecture.

The level and rigor of decomposition needed varies from enterprise to enterprise, as well as within an enterprise, and the architect should consider the enterprise's goals, objectives, scope, and purpose of the Enterprise Architecture effort to determine the level of decomposition.

4.3.1.2 Identify Required Catalogs of Business Building Blocks

Catalogs capture inventories of the core assets of the business. Catalogs are hierarchical in nature and capture the decomposition of a building block and also decompositions across related building blocks (e.g., organization/actor).

Catalogs form the raw material for development of matrices and views and also act as a key resource for managing the business and IT capability.

The TOGAF Standard — Architecture Content contains a detailed description of catalogs which should be considered for development within a Business Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

4.3.1.3 Identify Required Matrices

Matrices show the core relationships between related model entities.

Matrices form the raw material for development of views and also act as a key resource for impact assessment, carried out as a part of gap analysis.

The TOGAF Standard — Architecture Content contains a detailed description of matrices which should be considered for development within a Business Architecture, describing them in detail

and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

4.3.1.4 *Identify Required Diagrams*

Diagrams present the Business Architecture information from a set of different perspectives (viewpoints) according to the requirements of the stakeholders.

The TOGAF Standard — Architecture Content contains a detailed description of diagrams which should be considered for development within a Business Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

4.3.1.5 *Identify Types of Requirement to be Collected*

Once the Business Architecture catalogs, matrices, and diagrams have been developed, architecture modeling is completed by formalizing the business-focused requirements for implementing the Target Architecture.

These requirements may:

- Relate to the business domain
- Provide requirements input into the Data, Application, and Technology Architectures
- Provide detailed guidance to be reflected during design and implementation to ensure that the solution addresses the original architecture requirements

Within this step, the architect should identify requirements that should be met by the architecture (see [Section 13.5.2](#)).

In many cases, the Architecture Definition will not be intended to give detailed or comprehensive requirements for a solution (as these can be better addressed through general requirements management discipline). The expected scope of requirements content should be established during the Architecture Vision phase and documented in the approved Statement of Architecture Work.

Any requirement, or change in requirement, that is outside of the scope defined in the Statement of Architecture Work must be submitted to the Requirements Repository for management through the governed Requirements Management process.

4.3.2 **Develop Baseline Business Architecture Description**

Develop a Baseline Description of the existing Business Architecture, to the extent necessary to support the Target Business Architecture. The scope and level of detail to be defined will depend on the extent to which existing business elements are likely to be carried over into the Target Business Architecture, and on whether Architecture Descriptions exist, as described in [Section 4.5](#). To the extent possible, identify the relevant Business Architecture building blocks, drawing on the Architecture Repository (see the TOGAF Standard — Architecture Content).

Where new architecture models need to be developed to satisfy stakeholder concerns, use the models identified within Step 1 as a guideline for creating new architecture content to describe the Baseline Architecture.

4.3.3 Develop Target Business Architecture Description

Develop a Target Description for the Business Architecture, to the extent necessary to support the Architecture Vision. The scope and level of detail to be defined will depend on the relevance of the business elements to attaining the Target Architecture Vision, and on whether architectural descriptions exist. To the extent possible, identify the relevant Business Architecture building blocks, drawing on the Architecture Repository (see the TOGAF Standard — Architecture Content).

Where new architecture models need to be developed to satisfy stakeholder concerns, use the models identified within Step 1 as a guideline for creating new architecture content to describe the Target Architecture.

If appropriate, investigate different Target Architecture alternatives and discuss these with stakeholders using the Architecture Alternatives and Trade-offs technique (see the TOGAF Standard — ADM Techniques).

4.3.4 Perform Gap Analysis

Verify the architecture models for internal consistency and accuracy:

- Perform trade-off analysis to resolve conflicts (if any) among the different views
- Validate that the models support the principles, objectives, and constraints
- Note changes to the viewpoint represented in the selected models from the Architecture Repository, and document
- Test architecture models for completeness against requirements

Identify gaps between the baseline and target, using the gap analysis technique, as described in the TOGAF Standard — Applying the ADM.

4.3.5 Define Candidate Roadmap Components

Following the creation of a Baseline Architecture, Target Architecture, and gap analysis results, a business roadmap is required to prioritize activities over the coming phases.

This initial Business Architecture Roadmap will be used as raw material to support more detailed definition of a consolidated, cross-discipline roadmap within the Opportunities & Solutions phase.

4.3.6 Resolve Impacts Across the Architecture Landscape

Once the Business Architecture is finalized, it is necessary to understand any wider impacts or implications.

At this stage, other architecture artifacts in the Architecture Landscape should be examined to identify:

- Does this Business Architecture create an impact on any pre-existing architectures?

- Have recent changes been made that impact on the Business Architecture?
- Are there any opportunities to leverage work from this Business Architecture in other areas of the organization?
- Does this Business Architecture impact other projects (including those planned as well as those currently in progress)?
- Will this Business Architecture be impacted by other projects (including those planned as well as those currently in progress)?

4.3.7 Conduct Formal Stakeholder Review

Check the original motivation for the architecture project and the Statement of Architecture Work against the proposed Business Architecture, asking if it is fit for the purpose of supporting subsequent work in the other architecture domains. Refine the proposed Business Architecture only if necessary.

4.3.8 Finalize the Business Architecture

- Select standards for each of the building blocks, re-using as much as possible from the reference models selected from the Architecture Repository
- Fully document each building block
- Conduct a final cross-check of overall architecture against business goals; document the rationale for building block decisions in the architecture document
- Document the final requirements traceability report
- Document the final mapping of the architecture within the Architecture Repository; from the selected building blocks, identify those that might be re-used (working practices, roles, business relationships, job descriptions, etc.), and publish via the Architecture Repository
- Finalize all the work products, such as gap analysis results

4.3.9 Create/Update the Architecture Definition Document

- Document the rationale for building block decisions in the Architecture Definition Document
- Prepare the appropriate business sections of the Architecture Definition Document related to the intended scope and use of the architecture

If appropriate, use reports and/or graphics generated by modeling tools to demonstrate key views of the architecture. Route the document for review by relevant stakeholders, and incorporate feedback.

4.4 Outputs

The outputs of Phase B may include, but are not restricted to:

- Refined and updated versions of the Architecture Vision phase deliverables, where applicable, including:
 - Statement of Architecture Work (see the TOGAF Standard — Architecture Content), updated if necessary
 - Validated business principles, business goals, and business drivers (see the TOGAF Standard — Architecture Content), updated if necessary
 - Architecture Principles (see the TOGAF Standard — Architecture Content)
- Draft Architecture Definition Document (see the TOGAF Standard — Architecture Content), including:
 - Baseline Business Architecture, Approved, if appropriate
 - Target Business Architecture, Approved, including:
 - Organization structure — identifying business locations and relating them to organizational units
 - Business goals and objectives — for the enterprise and each organizational unit
 - Business functions — a detailed, recursive step involving successive decomposition of major functional areas into sub-functions
 - Business capabilities — the abilities that a business needs to possess or exchange to achieve its goals and objectives
 - Business services — the services that support the business by encapsulating a unique "elements of business behavior"; a service offered external to the enterprise may be supported by business services
 - Products — output generated by the business to be offered to customers; products include materials and/or services
 - Business processes, including measures and deliverables
 - Business roles, including development and modification of skills requirements
 - Business data model
 - Correlation of organization/business functions and business capabilities — relate business capabilities to organizational units in the form of a matrix report
 - Views corresponding to the selected viewpoints addressing key stakeholder concerns
- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including such Business Architecture requirements as:
 - Gap analysis results
 - Technical requirements — identifying, categorizing, and prioritizing the implications for work in the remaining architecture domains; for example, by a dependency/priority matrix (e.g., guiding trade-off between speed of transaction processing and security); list the specific models that are expected to be produced (for example, expressed as

primitives of the Zachman Framework)

— Updated business requirements

- Business Architecture components of an Architecture Roadmap (see the TOGAF Standard — Architecture Content)

The TOGAF Standard — Architecture Content contains a detailed description of architectural artifacts which might be produced in this phase.

4.5 Approach

Business Architecture is a representation of holistic, multi-dimensional business views of capabilities, end-to-end value delivery, information, and organizational structure; and the relationships among these business views and strategies, products, policies, initiatives, and stakeholders.

Business Architecture relates business elements to business goals and elements of other domains.

4.5.1 General

A knowledge of the Business Architecture is a prerequisite for architecture work in any other domain (Data, Application, Technology), and is therefore the first architecture activity that needs to be undertaken, if not catered for already in other organizational processes (enterprise planning, strategic business planning, business process re-engineering, etc.).

In practical terms, the Business Architecture is also often necessary as a means of demonstrating the business value of subsequent architecture work to key stakeholders, and the return on investment to those stakeholders from supporting and participating in the subsequent work.

The scope of work in Phase B is primarily determined by the Architecture Vision as set out in Phase A. The business strategy defines the goals and drivers and the metrics for success, but not necessarily how to get there. That is the role of the Business Architecture, defined in detail in Phase B.

This will depend to a large extent on the enterprise environment. In some cases, key elements of the Business Architecture may be done in other activities; for example, the enterprise mission, vision, strategy, and goals may be documented as part of some wider business strategy or enterprise planning activity that has its own lifecycle within the enterprise.

In such cases, there may be a need to verify and update the currently documented business strategy and plans, and/or to bridge between high-level business drivers, business strategy, and goals on the one hand, and the specific business requirements that are relevant to this architecture development effort.

In other cases, little or no Business Architecture work may have been done to date. In such cases, there will be a need for the architecture team to research, verify, and gain buy-in to the key business objectives and processes that the architecture is to support. This may be done as a free-standing exercise, either preceding architecture development, or as part of Phase A.

In both of these cases, the business scenarios technique (see the TOGAF® Series Guide: Business Scenarios), or any other method that illuminates the key business requirements and

indicates the implied technical requirements for IT architecture, may be used.

A key objective is to re-use existing material as much as possible. In architecturally more mature environments, there will be existing Architecture Definitions, which (hopefully) will have been maintained since the last architecture development cycle. Where Architecture Descriptions exist, these can be used as a starting point, and verified and updated if necessary; see the TOGAF Standard — Architecture Content.

Gather and analyze only that information that allows informed decisions to be made relevant to the scope of this architecture effort. If this effort is focused on the definition of (possibly new) business processes, then Phase B will necessarily involve a lot of detailed work. If the focus is more on the Target Architectures in other domains (data/information, application systems, infrastructure) to support an essentially existing Business Architecture, then it is important to build a complete picture in Phase B without going into unnecessary detail.

4.5.2 Developing the Baseline Description

If an enterprise has existing Architecture Descriptions, they should be used as the basis for the Baseline Description. This input may have been used already in Phase A in developing the Architecture Vision, such as the business capability map or a core set of value streams as introduced in [Section 3.5.2](#), and may be sufficient in itself for this baseline.

The reasons to update these materials include having a missing business capability, a new value stream, or changed organizational unit that has not previously been assessed within the scope of the Enterprise Architecture project. [Section 4.5.3](#) through [Section 4.5.6](#) address the use of core Business Architecture methods to model the Business Architecture driven by the strategy scope from Phase A. Note that putting these methods into action to drive a focus and target state for later architecture work does not mean that the fundamental frameworks from Phase A, such as a common enterprise business capability map, will necessarily change but rather that they are applied in a manner driven by the scope and needs of the specific Enterprise Architecture project.

If no Architecture Descriptions exist, information should be gathered and Business Architecture models developed.

Whatever the scope of the specific project, it is important to determine whether it is the fundamental view of the business that is changing or the usage of those views to determine scope, priorities, and relationships for the specific project in relation to the rest of the enterprise.

4.5.3 Applying Business Capabilities

The business capability map found or developed in the Architecture Vision phase provides a self-contained view of the business that is independent of the current organizational structure, business processes, information systems and applications, and the rest of the product or service portfolio. Those business capabilities should be mapped back to the organizational units, value streams, information systems, and strategic plans within the scope of the Enterprise Architecture project. This relationship mapping provides greater insight into the alignment and optimization of each of those domains (see Relationship Mapping in TOGAF® Series Guide: Business Capabilities).

Another common analysis technique involves heat mapping, which can be used to show a range of different perspectives on the same set of core business capabilities. These include maturity,

effectiveness, performance, and the value or cost of each capability to the business. Different attributes determine the colors of each capability on the business capability map (see Heat Mapping in TOGAF® Series Guide: Business Capabilities).

For example, a business capability maturity heat map shows the desired maturity as green for a specific capability, one level down as yellow, and two or more levels down as red. Other colors may indicate status, such as purple denoting a capability that does not exist yet in the company but is desired, or perhaps as a capability that is over-funded and has more resources than necessary. This gap analysis is directly tied to the Enterprise Architecture project underway; a gap is only relevant in the context of the business need and provides focus for more mapping in this phase or priorities for later architecture phases.

4.5.4 Applying Value Streams

Value streams provide valuable stakeholder context into why the organization needs business capabilities, while business capabilities provide what the organization needs for a particular value stage to be successful.

Start with the initial set of value stream models for the business documented in the Architecture Vision phase. Within the scope of the specific Enterprise Architecture project, if sufficiently larger in breadth, there may be a need for new value streams not already in the repository.

A new or existing value stream can be analyzed within the scope of the project through heat mapping (by value stream stage) or by developing use-cases around a complete definition of the value stream (see Baseline Example in the TOGAF® Series Guide: Value Streams). A project might focus on specific stakeholders, one element of business value, or stress some stages over others to develop better requirements for solutions in later phases.

The most substantive benefits come from mapping relationships between the stages in a value stream to business capabilities, then performing a gap analysis for capabilities (such as heat mapping) in the context of the business value achieved by the value stream for a specific stakeholder (see Mapping Value Streams to Business Capabilities in the TOGAF® Series Guide: Value Streams).

4.5.5 Applying the Organization Map

An organization map shows the key organizational units, partners, and stakeholder groups that make up the enterprise ecosystem. The map should also depict the working relationship between those entities, as distinct from an organizational chart that only shows hierarchical reporting relationships. The map is typically depicted as a network or web of relationships and interactions between the various business entities (see *Organigraphs: Drawing How Companies Really Work*, by Mintzberg and Van der Heyden, 1999).

The business unit is the main concept used to establish organization maps. In keeping with the relatively unconstrained view of what constitutes as enterprise, the enterprise may be one business unit for the project underway, may include all business units, or also include third parties or other stakeholder groups. The interpretation depends on the scope of the architecture effort.

This map is a key element of Business Architecture because it provides the organizational context for the whole Enterprise Architecture effort. While capability mapping exposes what a business does and value stream mapping exposes how it delivers value to specific stakeholders,

the organization map identifies the business units or third parties that possess or use those capabilities and which participate in the value streams.

Taken together with the methods in [Section 4.5.3](#), [Section 4.5.4](#), and the associated Guides, the organization map provides an understanding of which business units to involve in the architecture effort, who and when to talk about a given requirement, and how to measure the impact of various decisions.

For further guidance on organization maps, see the TOGAF® Series Guide: Organization Mapping.

4.5.6 Applying Information Maps

Characterizing information in the Business Architecture phase starts with the elements that matter most to the business, such as product, customer, factory, etc. With a list of these key elements, a cross-discipline team can list and map the information that matters most and how it is described using business terms. Domains of information can be derived from groupings or categories that the business terms fall into logically. These domains are a good place to start the information map for completeness and highest value before proceeding later to the details of data types.

Relationships among the information domains can then be added to the map as the next level of understanding for a good baseline information map. The most significant benefit then comes with building matrices between information and business capabilities. The linkage between the information that matters most to the business and the business capabilities that describe the ability to apply that information to create value is a key aspect of Business Architecture. These information maps and relationships to business capabilities will then apply in later architecture phases on data characterization, applications, and infrastructure.

For further guidance on information maps, see the TOGAF® Series Guide: Information Mapping.

4.5.7 Applying Modeling Techniques

The modeling and mapping techniques provided here are extensions that implement the business capabilities, value streams, and organization maps described above in Phase B into the practices of the business. They expand the operating model, which is a representation for how an organization operates across a range of domains in order to accomplish its purpose (see *A Method for Identifying Process Re-Use Opportunities to Enhance the Operating Model*, M. de Vries et al.).

In addition to the techniques described above (capability maps, value streams, and organization maps), a variety of other modeling techniques may be employed, if deemed appropriate. For example:

- **Activity Models** (also called **Business Process Models**) describe the enterprise's business activities, the data and/or information exchanged between activities (internal exchanges), and the data and/or information exchanged with other activities that are outside the scope of the model (external exchanges)

Activity models are hierarchical in nature. They capture the activities performed in a business process, and the Inputs, Controls, Outputs, and Mechanisms/Resources (ICOMs) of those activities. Activity models can be annotated with explicit statements of business rules, which represent relationships among the ICOMs. For example, a business rule can

specify who can do what under specified conditions, the combination of inputs and controls needed, and the resulting outputs. One technique for creating activity models is the Integrated Computer Aided Manufacturing (ICAM) DEFinition (IDEF) modeling technique.

The Object Management Group (OMG) has developed the Business Process Modeling Notation™ (BPMN™), a standard for business process modeling that includes a language with which to specify business processes, their tasks/steps, and the documents produced.

- **Use-Case Models** describe the business process of an enterprise in terms of use-cases and actors corresponding to business processes and organizational participants (people, organizations, etc.)

The use-case model is described in use-case diagrams and use-case specifications.

- **Logical Data Model (or Class Model)**

Logical data models describe the entities, their attributes, and the acceptable values for these attributes as well as the relationships between the various entities. Particularly useful is the "is-a" relationship for analyses. For example, if a sedan is a type of car which is a type of vehicle, then a general business rule for all vehicles can be written once and automatically picked up for all types of cars and trucks.

A class model expands the logical data model to include behaviors by including services (methods) unique to the entity (now called a class). Normally applications and information architects will use the class diagram, and Business Architects will use the logical data model.

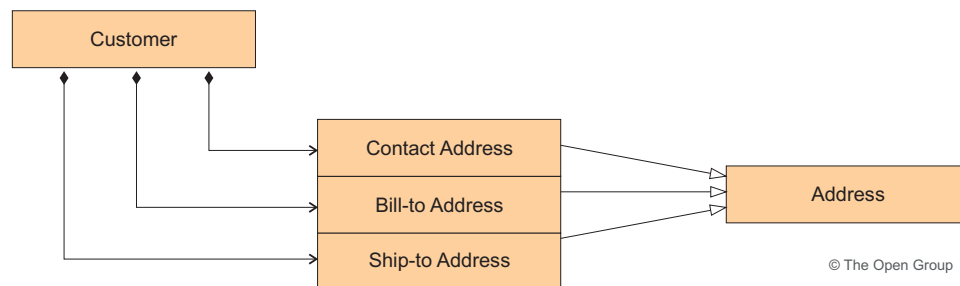


Figure 4-2 UML Business Class Diagram

All three types of model above can be represented in the Unified Modeling Language™ (UML®), and a variety of tools exist for generating such models.

Certain industry sectors have modeling techniques specific to the sector concerned.

4.5.8 Architecture Repository

As part of Phase B, the architecture team will need to consider what relevant Business Architecture resources are available from the Architecture Repository (see the TOGAF Standard — Architecture Content), in particular:

- Industry reference models relevant to the organization's industry sector
- Enterprise-specific Business Architecture views (capability maps, value stream maps, organization maps, etc.)
- Enterprise-specific building blocks (process components, business rules, job descriptions, etc.)
- Applicable standards

Chapter 5: Phase C: Information Systems Architectures

This chapter describes the Information Systems Architectures for an architecture project, including the development of Data and Application Architectures.

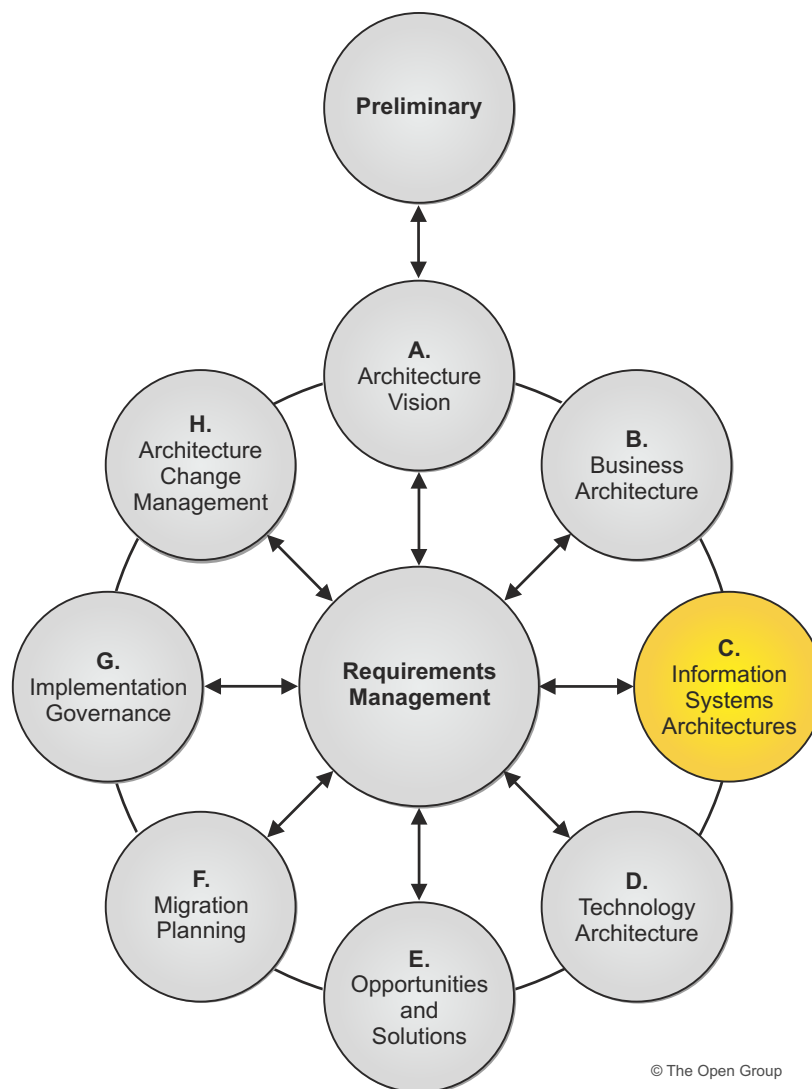


Figure 5-1 Phase C: Information Systems Architectures

5.1 Objectives

The objectives of Phase C are to:

- Develop the Target Information Systems Architectures, describing how the enterprise's Information Systems Architecture will enable the Business Architecture and the Architecture Vision, in a way that addresses the Statement of Architecture Work and stakeholder concerns
- Identify candidate Architecture Roadmap components based upon gaps between the Baseline and Target Information Systems (Data and Application) Architectures

5.2 Approach

Phase C involves some combination of Data and Application Architecture, in either order. Advocates exist for both sequences. For example, Steven Spewak's *Enterprise Architecture Planning (EAP)* recommends a data-driven approach.

On the other hand, major applications systems — such as those for Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), etc. — often provide a combination of technology infrastructure and business application logic. Some organizations take an application-driven approach, whereby they recognize certain key applications as forming the core underpinning of the mission-critical business processes, and take the implementation and integration of those core applications as the primary focus of their architecture effort (the integration issues often constituting a major challenge).

Detailed descriptions for Phase C are given separately for each architecture domain:

- Phase C: Information Systems Architectures — Data Architecture (see [Chapter 6](#))
- Phase C: Information Systems Architectures — Application Architecture (see [Chapter 7](#))

Chapter 6: Phase C: Information Systems Architectures — Data Architecture

This chapter describes the Data Architecture part of Phase C.

6.1 Objectives

The objectives of the Data Architecture part of Phase C are to:

- Develop the Target Data Architecture that enables the Business Architecture and the Architecture Vision, in a way that addresses the Statement of Architecture Work and stakeholder concerns
- Identify candidate Architecture Roadmap components based upon gaps between the Baseline and Target Data Architectures

6.2 Inputs

This section defines the inputs to Phase C (Data Architecture).

6.2.1 Reference Materials External to the Enterprise

- Architecture reference materials (see the TOGAF Standard — Architecture Content)
- TOGAF® Series Guide: Information Architecture — Customer Master Data Management
- The Open Group Guide: Information Architecture: Business Intelligence & Analytics and Metadata Management Reference Models

6.2.2 Non-Architectural Inputs

- Request for Architecture Work (see the TOGAF Standard — Architecture Content)
- Capability Assessment (see the TOGAF Standard — Architecture Content)
- Communications Plan (see the TOGAF Standard — Architecture Content)

6.2.3 Architectural Inputs

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Budget requirements
 - Governance and support strategy
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)
 - Configured and deployed tools
- Data principles (see the TOGAF Standard — ADM Techniques), if existing
- Statement of Architecture Work (see the TOGAF Standard — Architecture Content)
- Architecture Vision (see the TOGAF Standard — Architecture Content)
- Architecture Repository (see the TOGAF Standard — Architecture Content), including:
 - Re-usable building blocks (in particular, definitions of current data)
 - Publicly available reference models
 - Organization-specific reference models
 - Organization standards
- Draft Architecture Definition Document, which may include Baseline and/or Target Architectures of any architecture domain
- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including:
 - Gap analysis results (from Business Architecture)
 - Relevant technical requirements that will apply to this phase
- Business Architecture components of an Architecture Roadmap (see the TOGAF Standard — Architecture Content)

6.3 Steps

The level of detail addressed in Phase C will depend on the scope and goals of the overall architecture effort.

New data building blocks being introduced as part of this effort will need to be defined in detail during Phase C. Existing data building blocks to be carried over and supported in the target environment may already have been adequately defined in previous architectural work; but, if not, they too will need to be defined in Phase C.

The order of the steps in this phase as well as the time at which they are formally started and completed should be adapted to the situation at hand in accordance with the established Architecture Governance. In particular, determine whether in this situation it is appropriate to conduct Baseline Description or Target Architecture development first, as described in the TOGAF Standard — Applying the ADM.

All activities that have been initiated in these steps should be closed during the Finalize the Data Architecture step (see [Section 6.3.8](#)). The documentation generated from these steps must be formally published in the Create/Update the Architecture Definition Document step (see [Section 6.3.9](#)).

The steps in Phase C (Data Architecture) are as follows:

- Select reference models, viewpoints, and tools (see [Section 6.3.1](#))
- Develop Baseline Data Architecture Description (see [Section 6.3.2](#))
- Develop Target Data Architecture Description (see [Section 6.3.3](#))
- Perform gap analysis (see [Section 6.3.4](#))
- Define candidate roadmap components (see [Section 6.3.5](#))
- Resolve impacts across the Architecture Landscape (see [Section 6.3.6](#))
- Conduct formal stakeholder review (see [Section 6.3.7](#))
- Finalize the Data Architecture (see [Section 6.3.8](#))
- Create/update the Architecture Definition Document (see [Section 6.3.9](#))

6.3.1 Select Reference Models, Viewpoints, and Tools

Review and validate (or generate, if necessary) the set of data principles. These will normally form part of an overarching set of Architecture Principles. Guidelines for developing and applying principles, and a sample set of data principles, are given in the TOGAF Standard — ADM Techniques.

Select relevant Data Architecture resources (reference models, patterns, etc.) on the basis of the business drivers, stakeholders, concerns, and Business Architecture.

Select relevant Data Architecture viewpoints (for example, stakeholders of the data — regulatory bodies, users, generators, subjects, auditors, etc.; various time dimensions — real-time, reporting period, event-driven, etc.; locations; business processes); i.e., those that will enable the architect to demonstrate how the stakeholder concerns are being addressed in the Data Architecture.

Identify appropriate tools and techniques (including forms) to be used for data capture,

modeling, and analysis, in association with the selected viewpoints. Depending on the degree of sophistication warranted, these may comprise simple documents or spreadsheets, or more sophisticated modeling tools and techniques such as data management models, data models, etc.

Examples of data modeling techniques are:

- Entity relationship diagram
- Class diagram

Further guidance on Information Architecture reference models can be found in the following documents:

- TOGAF® Series Guide: Information Architecture — Customer Master Data Management
- The Open Group Guide: Information Architecture: Business Intelligence & Analytics and Metadata Management Reference Models

6.3.1.1 Determine Overall Modeling Process

For each viewpoint, select the models needed to support the specific view required, using the selected tool or method.

Ensure that all stakeholder concerns are covered. If they are not, create new models to address concerns not covered, or augment existing models (see above).

The recommended process for developing a Data Architecture is as follows:

- Collect data-related models from existing Business Architecture and Application Architecture materials
- Rationalize data requirements and align with any existing enterprise data catalogs and models; this allows the development of a data inventory and entity relationship
- Update and develop matrices across the architecture by relating data to business service, business capability, business function, access rights, and application
- Elaborate Data Architecture views by examining how data is created, distributed, migrated, secured, and archived

6.3.1.2 Identify Required Catalogs of Data Building Blocks

Descriptions of data may be captured as a catalog showing decomposition across related model entities (e.g., data entity -> logical data component -> physical data component).

During the Business Architecture phase, a Business Service/Information diagram was created showing the key data entities required by the main business services. This is a prerequisite to successful Data Architecture activities.

Using the traceability from business function/business capability to application and data entity, it is possible to create an inventory of the data needed to support the Architecture Vision.

Once the data requirements are consolidated in a single location, it is possible to refine the data inventory to achieve semantic consistency and to remove gaps and overlaps.

The TOGAF Standard — Architecture Content contains a detailed description of catalogs which should be considered for development within a Data Architecture, describing them in detail and

relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

6.3.1.3 *Identify Required Matrices*

At this stage, an entity to applications matrix could be produced to validate this mapping. How data is created, maintained, transformed, and passed to other applications, or used by other applications, will now start to be understood. Obvious gaps such as entities that never seem to be created by an application or data created but never used, need to be noted for later gap analysis.

The rationalized data inventory can be used to update and refine the architectural diagrams of how data relates to other aspects of the architecture.

Once these updates have been made, it may be appropriate to drop into a short iteration of the Application Architecture to resolve the changes identified.

The TOGAF Standard — Architecture Content contains a detailed description of matrices which should be considered for development within a Data Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

6.3.1.4 *Identify Required Diagrams*

Diagrams present the Data Architecture information from a set of different perspectives (viewpoints) according to the requirements of the stakeholders.

Once the data entities have been refined, a diagram of the relationships between entities and their attributes can be produced.

It is important to note at this stage that information may be a mixture of enterprise-level data (from system service providers and package vendor information) and local-level data held in personal databases and spreadsheets.

The level of detail modeled needs to be carefully assessed. Some physical system data models will exist down to a very detailed level; others will only have core entities modeled. Not all data models will have been kept up-to-date as applications were modified and extended over time. It is important to achieve a balance in the level of detail provided (e.g., the reproduction of existing detailed system physical data schemas or the presentation of high-level process maps and data requirements highlight the two extreme views).

The TOGAF Standard — Architecture Content contains a detailed description of diagrams which should be considered for development within a Data Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

6.3.1.5 *Identify Types of Requirement to be Collected*

Once the Data Architecture catalogs, matrices, and diagrams have been developed, architecture modeling is completed by formalizing the data-focused requirements for implementing the Target Architecture.

These requirements may:

- Relate to the data domain

- Provide requirements input into the Application and Technology Architectures
- Provide detailed guidance to be reflected during design and implementation to ensure that the solution addresses the original architecture requirements

Within this step, the architect should identify requirements that should be met by the architecture (see [Section 13.5.2](#)).

6.3.2 Develop Baseline Data Architecture Description

Develop a Baseline Description of the existing Data Architecture, to the extent necessary to support the Target Data Architecture. The scope and level of detail to be defined will depend on the extent to which existing data elements are likely to be carried over into the Target Data Architecture, and on whether architectural descriptions exist, as described in [Section 6.5](#). To the extent possible, identify the relevant Data Architecture building blocks, drawing on the Architecture Repository (see the TOGAF Standard — Architecture Content).

Where new architecture models need to be developed to satisfy stakeholder concerns, use the models identified within Step 1 as a guideline for creating new architecture content to describe the Baseline Architecture.

6.3.3 Develop Target Data Architecture Description

Develop a Target Description for the Data Architecture, to the extent necessary to support the Architecture Vision and Target Business Architecture. The scope and level of detail to be defined will depend on the relevance of the data elements to attaining the Target Architecture, and on whether architectural descriptions exist. To the extent possible, identify the relevant Data Architecture building blocks, drawing on the Architecture Repository (see TOGAF Standard — Architecture Content).

Where new architecture models need to be developed to satisfy stakeholder concerns, use the models identified within Step 1 as a guideline for creating new architecture content to describe the Target Architecture.

If appropriate, investigate different Target Architecture alternatives and discuss these with stakeholders using the Architecture Alternatives and Trade-offs technique (see the TOGAF Standard — ADM Techniques).

6.3.4 Perform Gap Analysis

Verify the architecture models for internal consistency and accuracy:

- Perform trade-off analysis to resolve conflicts (if any) among the different views
- Validate that the models support the principles, objectives, and constraints
- Note changes to the viewpoint represented in the selected models from the Architecture Repository, and document
- Test architecture models for completeness against requirements

Identify gaps between the Baseline and Target, using the gap analysis technique as described in the TOGAF Standard — ADM Techniques.

6.3.5 Define Candidate Roadmap Components

Following the creation of a Baseline Architecture, Target Architecture, and gap analysis, a data roadmap is required to prioritize activities over the coming phases.

This initial Data Architecture roadmap will be used as raw material to support more detailed definition of a consolidated, cross-discipline roadmap within the Opportunities & Solutions phase.

6.3.6 Resolve Impacts Across the Architecture Landscape

Once the Data Architecture is finalized, it is necessary to understand any wider impacts or implications.

At this stage, other architecture artifacts in the Architecture Landscape should be examined to identify:

- Does this Data Architecture create an impact on any pre-existing architectures?
- Have recent changes been made that impact the Data Architecture?
- Are there any opportunities to leverage work from this Data Architecture in other areas of the organization?
- Does this Data Architecture impact other projects (including those planned as well as those currently in progress)?
- Will this Data Architecture be impacted by other projects (including those planned as well as those currently in progress)?

6.3.7 Conduct Formal Stakeholder Review

Check the original motivation for the architecture project and the Statement of Architecture Work against the proposed Data Architecture. Conduct an impact analysis to identify any areas where the Business and Application Architectures (e.g., business practices) may need to change to cater for changes in the Data Architecture (for example, changes to forms or procedures, applications, or database systems).

If the impact is significant, this may warrant the Business and Application Architectures being revisited.

Identify any areas where the Application Architecture (if generated at this point) may need to change to cater for changes in the Data Architecture (or to identify constraints on the Application Architecture about to be designed).

If the impact is significant, it may be appropriate to drop into a short iteration of the Application Architecture at this point.

Identify any constraints on the Technology Architecture about to be designed, refining the proposed Data Architecture only if necessary.

6.3.8 Finalize the Data Architecture

- Select standards for each of the building blocks, re-using as much as possible from the reference models selected from the Architecture Repository
- Fully document each building block
- Conduct a final cross-check of overall architecture against business requirements; document the rationale for building block decisions in the architecture document
- Document the final requirements traceability report
- Document the final mapping of the architecture within the Architecture Repository; from the selected building blocks, identify those that might be re-used, and publish via the Architecture Repository
- Finalize all the work products, such as gap analysis

6.3.9 Create/Update the Architecture Definition Document

Document the rationale for building block decisions in the Architecture Definition Document.

Prepare the Data Architecture sections of the Architecture Definition Document, comprising some or all of:

- Business data model
- Logical data model
- Data management process model
- Data Entity/Business Function matrix
- Data interoperability requirements (e.g., XML schema, security policies)
- If appropriate, use reports and/or graphics generated by modeling tools to demonstrate key views of the architecture; route the document for review by relevant stakeholders, and incorporate feedback

6.4 Outputs

The outputs of Phase C (Data Architecture) may include, but are not restricted to:

- Refined and updated versions of the Architecture Vision phase deliverables, where applicable:
 - Statement of Architecture Work (see the TOGAF Standard — Architecture Content), updated if necessary
 - Validated data principles (see the TOGAF Standard — ADM Techniques), or new data principles (if generated here)
- Draft Architecture Definition Document (see the TOGAF Standard — Architecture Content), including:
 - Baseline Data Architecture, Approved, if appropriate

- Target Data Architecture, Approved, including:
 - Business data model
 - Logical data model
 - Data management process models
 - Data Entity/Business Function matrix
- Views corresponding to the selected viewpoints addressing key stakeholder concerns
- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including such Data Architecture requirements as:
 - Gap analysis results
 - Data interoperability requirements
 - Relevant technical requirements that will apply to this evolution of the architecture development cycle
 - Constraints on the Technology Architecture about to be designed
 - Updated business requirements, if appropriate
 - Updated application requirements, if appropriate
- Data Architecture components of an Architecture Roadmap (see the TOGAF Standard — Architecture Content)

The TOGAF Standard — Architecture Content contains a detailed description of architectural artifacts which might be produced in this phase.

6.5 Approach

6.5.1 Data Structure

A Data Architecture should be able to handle:

- Data at rest — data in stores
- Data in motion — data in transactions or services/APIs
- Data in use — data at the border of the application (e.g., GUI)
- Open data — data that the organization provides for public usage and which it is voluntarily or legally required to provide

Different alternate ways of working with these types of Data Architecture will be added.

Data Architecture is created by using three metamodel entities: data entity, logical data component, and physical data component.

Data entities can be used to create conceptual data models to help the IT developers understand the concepts they will be dealing with. Often the entity relationship models also contain some requirements on the relations (e.g., a customer can only have one address).

Logical data components can be used to create logical data models. Often it is important for the IT area to have a clear view of all data that is used in the IT environment. The logical data model is often used as a requirement on the data stored in applications (at rest), data moved between

applications (in motion), or data at the user interface of applications (data in use).

Physical data components are clusters of logical data components that have been implemented by some earlier project (links to, for example, XML message, database schemas) or requirements for new implementation projects.

All three data entities can be used in data exchange models for data passed between/into/out of IS services, logical application components, or physical application components.

All data entities can have quality attributes for specific situations.

6.5.2 Key Considerations for Data Architecture

6.5.2.1 Data Management

When an enterprise has chosen to undertake large-scale architectural transformation, it is important to understand and address data management issues. A structured and comprehensive approach to data management enables the effective use of data to capitalize on its competitive advantages.

Considerations include:

- A clear definition of which application components in the landscape will serve as the system of record or reference for enterprise master data
- Will there be an enterprise-wide standard that all application components, including software packages, need to adopt?
(In the main, packages can be prescriptive about the data models and may not be flexible.)
- Clearly understand how data entities are utilized by business capabilities, business functions, processes, and business and application services
- Clearly understand how and where enterprise data entities are created, stored, transported, and reported
- What is the level and complexity of data transformations required to support the information exchange needs between applications?
- What will be the requirement for software in supporting data integration with the enterprise's customers and suppliers (e.g., use of Extract, Transform, Load (ETL) tools during data migration, data profiling tools to evaluate data quality, etc.)?

More guidance on data management can be found in the TOGAF® Series Guide: Information Architecture — Customer Master Data Management.

6.5.2.2 Data Migration

When an existing application is replaced, there will be a critical need to migrate data (master, transactional, and reference) to the new application. The Data Architecture should identify data migration requirements and also provide indicators as to the level of transformation, weeding, and cleansing that will be required to present data in a format that meets the requirements and constraints of the target application. The objective being that the target application has quality data when it is populated. Another key consideration is to ensure that an enterprise-wide common data definition is established to support the transformation.

6.5.2.3 Data Governance

Data governance considerations ensure that the enterprise has the necessary dimensions in place to enable the transformation, as follows:

- **Structure:** this dimension pertains to whether the enterprise has the necessary organizational structure and the standards bodies to manage data entity aspects of the transformation
- **Management System:** here enterprises should have the necessary management system and data-related programs to manage the governance aspects of data entities throughout its lifecycle
- **People:** this dimension addresses what data-related skills and roles the enterprise requires for the transformation

If the enterprise lacks such resources and skills, the enterprise should consider either acquiring those critical skills or training existing internal resources to meet the requirements through a well-defined learning program.

6.5.3 Architecture Repository

As part of this phase, the architecture team will need to consider what relevant Data Architecture resources are available in the organization's Architecture Repository (see the TOGAF Standard — Architecture Content); in particular, generic data models relevant to the organization's industry "vertical" sector.

Chapter 7: Phase C: Information Systems Architectures — Application Architecture

This chapter describes the Application Architecture part of Phase C.

7.1 Objectives

The objectives of the Application Architecture part of Phase C are to:

- Develop the Target Application Architecture that enables the Business Architecture and the Architecture Vision, in a way that addresses the Statement of Architecture Work and stakeholder concerns
- Identify candidate Architecture Roadmap components based upon gaps between the Baseline and Target Application Architectures

7.2 Inputs

This section defines the inputs to Phase C (Application Architecture).

7.2.1 Reference Materials External to the Enterprise

- Architecture reference materials (see the TOGAF Standard — Architecture Content)

7.2.2 Non-Architectural Inputs

- Request for Architecture Work (see the TOGAF Standard — Architecture Content)
- Capability Assessment (see the TOGAF Standard — Architecture Content)
- Communications Plan (see the TOGAF Standard — Architecture Content)

7.2.3 Architectural Inputs

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach

- Roles and responsibilities for architecture team(s)
- Constraints on architecture work
- Budget requirements
- Governance and support strategy
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)
 - Configured and deployed tools
- Application principles (see the TOGAF Standard — ADM Techniques), if existing
- Statement of Architecture Work (see the TOGAF Standard — Architecture Content)
- Architecture Vision (see the TOGAF Standard — Architecture Content)
- Architecture Repository (see the TOGAF Standard — Architecture Content), including:
 - Re-usable building blocks
 - Publicly available reference models
 - Organization-specific reference models
 - Organization standards
- Draft Architecture Definition Document, which may include Baseline and/or Target Architectures of any architecture domain
- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including:
 - Gap analysis results (from Business Architecture and Data Architecture, if available)
 - Relevant technical requirements that will apply to this phase
- Business and Data Architecture components of an Architecture Roadmap, if available (see the TOGAF Standard — Architecture Content),

7.3 Steps

The level of detail addressed in Phase C will depend on the scope and goals of the overall architecture effort.

New application building blocks being introduced as part of this effort will need to be defined in detail during Phase C. Existing application building blocks to be carried over and supported in the target environment may already have been adequately defined in previous architectural work; but, if not, they too will need to be defined in Phase C.

The order of the steps in this phase as well as the time at which they are formally started and completed should be adapted to the situation at hand in accordance with the established Architecture Governance. In particular, determine whether in this situation it is appropriate to conduct Baseline Description or Target Architecture development first, as described in the TOGAF Standard — Applying the ADM.

All activities that have been initiated in these steps should be closed during the Finalize the Application Architecture step (see [Section 7.3.8](#)). The documentation generated from these steps must be formally published in the Create/Update the Architecture Definition Document step (see [Section 7.3.9](#)).

The steps in Phase C (Application Architecture) are as follows:

- Select reference models, viewpoints, and tools (see [Section 7.3.1](#))
- Develop Baseline Application Architecture Description (see [Section 7.3.2](#))
- Develop Target Application Architecture Description (see [Section 7.3.3](#))
- Perform gap analysis (see [Section 7.3.4](#))
- Define candidate roadmap components (see [Section 7.3.5](#))
- Resolve impacts across the Architecture Landscape (see [Section 7.3.6](#))
- Conduct formal stakeholder review (see [Section 7.3.7](#))
- Finalize the Application Architecture (see [Section 7.3.8](#))
- Create/Update the Architecture Definition Document (see [Section 7.3.9](#))

7.3.1 Select Reference Models, Viewpoints, and Tools

Review and validate (or generate, if necessary) the set of application principles. These will normally form part of an overarching set of Architecture Principles. Guidelines for developing and applying principles, and a sample set of application principles, are given in the TOGAF Standard — ADM Techniques.

Select relevant Application Architecture resources (reference models, patterns, etc.) from the Architecture Repository, on the basis of the business drivers, the stakeholders, and their concerns.

Select relevant Application Architecture viewpoints (for example, stakeholders of the applications — viewpoints relevant to functional and individual users of applications, etc.); i.e., those that will enable the architect to demonstrate how the stakeholder concerns are being addressed in the Application Architecture.

Identify appropriate tools and techniques to be used for capture, modeling, and analysis, in association with the selected viewpoints. Depending on the degree of sophistication warranted, these may comprise simple documents or spreadsheets, or more sophisticated modeling tools and techniques.

Consider using platform-independent descriptions of business logic. For example, the OMG Model-Driven Architecture[®] (MDA[®]) offers an approach to modeling Application Architectures that preserves the business logic from changes to the underlying platform and implementation technology.

7.3.1.1 Determine Overall Modeling Process

For each viewpoint, select the models needed to support the specific view required, using the selected tool or method.

Ensure that all stakeholder concerns are covered. If they are not, create new models to address concerns not covered, or augment existing models (see above).

The recommended process for developing an Application Architecture is as follows:

- Understand the list of applications or application components that are required, based on the baseline Application Portfolio, what the requirements are, and the Business Architecture scope
- Simplify complicated applications by decomposing them into two or more applications
- Ensure that the set of application definitions is internally consistent, by removing duplicate functionality as far as possible, and combining similar applications into one
- Identify logical applications and the most appropriate physical applications
- Develop matrices across the architecture by relating applications to business services, business capabilities, data, processes, etc.
- Elaborate a set of Application Architecture views by examining how the application will function, capturing integration, migration, development, and operational concerns

The level and rigor of decomposition needed varies from enterprise to enterprise, as well as within an enterprise, and the architect should consider the enterprise's goals, objectives, scope, and purpose of the Enterprise Architecture effort to determine the level of decomposition.

The level of granularity should be sufficient to enable identification of gaps and the scope of candidate work packages.

7.3.1.2 Identify Required Catalogs of Application Building Blocks

The organization's Application Portfolio is captured as a catalog within the Architecture Repository. Catalogs are hierarchical in nature and capture a decomposition of a metamodel entity and also decompositions across related model entities (e.g., logical application component → physical application component → application service).

Catalogs form the raw material for the development of matrices and diagrams and also act as a key resource for managing the business and IT capability.

The structure of catalogs is based on the attributes of metamodel entities, as defined in the TOGAF Standard — Architecture Content.

The TOGAF Standard — Architecture Content contains a detailed description of catalogs which should be considered for development within an Application Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

7.3.1.3 Identify Required Matrices

Matrices show the core relationships between related model entities.

Matrices form the raw material for development of diagrams and also act as a key resource for impact assessment.

Once the baseline Application Portfolio has been assembled, it is necessary to map the applications to their purpose in supporting the business. The initial mapping should focus on business services within the Business Architecture, as this is the level of granularity where architecturally significant decisions are most likely to be needed.

Once applications are mapped to business services, it will also be possible to make associations from applications to data, through the business-information diagrams developed during Business Architecture.

If readily available, baseline application data models may be used to validate the Business Architecture and also to identify which data is held locally and which is accessed remotely.

The Data Architecture phase will focus on these issues, so at this point it may be appropriate to drop into a short iteration of the Data Architecture if it is deemed to be valuable to the scope of the architecture engagement.

Using existing information in the baseline application catalog, the Application Architecture should identify user and organizational dependencies on applications. This activity will support future state planning by determining impacted user communities and also facilitating the grouping of applications by user type or user location.

A key user community to be specifically considered is the operational support organization. This activity should examine application dependencies on shared operations capabilities and produce a diagram on how each application is effectively operated and managed.

Specifically considering the needs of the operational community may identify requirements for new or extended governance capabilities and applications.

The TOGAF Standard — Architecture Content contains a detailed description of matrices which should be considered for development within an Application Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

7.3.1.4 Identify Required Diagrams

Diagrams present the Application Architecture information from a set of different perspectives (viewpoints) according to the requirements of the stakeholders. Once the desired functionality of an application is known, it is necessary to perform an internal assessment of how the application should be best structured to meet its requirements.

In the case of packaged applications, it is likely to be the case that the application supports a number of configuration options, add-on modules, or application services that may be applied to the solution. For custom developed applications, it is necessary to identify the high-level structure of the application in terms of modules or subsystems as a foundation to organize design activity.

The TOGAF Standard — Architecture Content contains a detailed description of diagrams which should be considered for development within an Application Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

7.3.1.5 Identify Types of Requirement to be Collected

Once the Application Architecture catalogs, matrices, and diagrams have been developed, architecture modeling is completed by formalizing the application-focused requirements for implementing the Target Architecture.

These requirements may:

- Relate to the application domain
- Provide requirements input into the Data and Technology Architectures
- Provide detailed guidance to be reflected during design and implementation to ensure that the solution addresses the original architecture requirements

Within this step, the architect should identify requirements that should be met by the architecture (see [Section 13.5.2](#)).

7.3.2 Develop Baseline Application Architecture Description

Develop a Baseline Description of the existing Application Architecture, to the extent necessary to support the Target Application Architecture. The scope and level of detail to be defined will depend on the extent to which existing applications are likely to be carried over into the Target Application Architecture, and on whether Architecture Descriptions exist, as described in [Section 7.5](#). To the extent possible, identify the relevant Application Architecture building blocks, drawing on the Architecture Repository (see the TOGAF Standard — Architecture Content). If not already existing within the Architecture Repository, define each application in line with the Application Portfolio catalog (see the TOGAF Standard — Architecture Content).

Where new architecture models need to be developed to satisfy stakeholder concerns, use the models identified within Step 1 as a guideline for creating new architecture content to describe the Baseline Architecture.

7.3.3 Develop Target Application Architecture Description

Develop a Target Description for the Application Architecture, to the extent necessary to support the Architecture Vision, Target Business Architecture, and Target Data Architecture. The scope and level of detail to be defined will depend on the relevance of the application elements to attaining the Target Architecture Vision, and on whether architectural descriptions exist. To the extent possible, identify the relevant Application Architecture building blocks, drawing on the Architecture Repository (see the TOGAF Standard — Architecture Content).

Where new architecture models need to be developed to satisfy stakeholder concerns, use the models identified within Step 1 as a guideline for creating new architecture content to describe the Target Architecture.

If appropriate, investigate different Target Architecture alternatives and discuss these with stakeholders using the Architecture Alternatives and Trade-offs technique (see the TOGAF Standard — ADM Techniques).

7.3.4 Perform Gap Analysis

Verify the architecture models for internal consistency and accuracy:

- Perform trade-off analysis to resolve conflicts (if any) among the different views
- Validate that the models support the principles, objectives, and constraints
- Note changes to the viewpoint represented in the selected models from the Architecture Repository, and document
- Test architecture models for completeness against requirements

Identify gaps between the baseline and target, using the gap analysis technique as described in the TOGAF Standard — ADM Techniques.

7.3.5 Define Candidate Roadmap Components

Following the creation of a Baseline Architecture, Target Architecture, and gap analysis, an application roadmap is required to prioritize activities over the coming phases.

This initial Application Architecture roadmap will be used as raw material to support more detailed definition of a consolidated, cross-discipline roadmap within the Opportunities & Solutions phase.

7.3.6 Resolve Impacts Across the Architecture Landscape

Once the Application Architecture is finalized, it is necessary to understand any wider impacts or implications.

At this stage, other architecture artifacts in the Architecture Landscape should be examined to identify:

- Does this Application Architecture create an impact on any pre-existing architectures?
- Have recent changes been made that impact the Application Architecture?
- Are there any opportunities to leverage work from this Application Architecture in other areas of the organization?
- Does this Application Architecture impact other projects (including those planned as well as those currently in progress)?
- Will this Application Architecture be impacted by other projects (including those planned as well as those currently in progress)?

7.3.7 Conduct Formal Stakeholder Review

Check the original motivation for the architecture project and the Statement of Architecture Work against the proposed Application Architecture. Conduct an impact analysis, to identify any areas where the Business and Data Architectures (e.g., business practices) may need to change to cater for changes in the Application Architecture (for example, changes to forms or procedures, applications, or database systems). If the impact is significant, this may warrant the Business and Data Architectures being revisited.

Identify any constraints on the Technology Architecture (especially the infrastructure) about to be designed.

7.3.8 Finalize the Application Architecture

- Select standards for each of the building blocks, re-using as much as possible from the reference models selected from the Architecture Repository
- Fully document each building block
- Conduct a final cross-check of overall architecture against business requirements; document the rationale for building block decisions in the architecture document
- Document the final requirements traceability report
- Document the final mapping of the architecture within the Architecture Repository; from the selected building blocks, identify those that might be re-used, and publish via the Architecture Repository
- Finalize all the work products, such as gap analysis

7.3.9 Create/Update the Architecture Definition Document

- Document the rationale for building block decisions in the Architecture Definition Document
- Prepare the Application Architecture sections of the Architecture Definition Document; if appropriate, use reports and/or graphics generated by modeling tools to demonstrate key views of the architecture; route the document for review by relevant stakeholders, and incorporate feedback

7.4 Outputs

The outputs of Phase C (Application Architecture) may include, but are not restricted to:

- Refined and updated versions of the Architecture Vision phase deliverables, where applicable:
 - Statement of Architecture Work (see the TOGAF Standard — Architecture Content), updated if necessary
 - Validated application principles, or new application principles (if generated here)

- Draft Architecture Definition Document (see the TOGAF Standard — Architecture Content), including:
 - Baseline Application Architecture, Approved, if appropriate
 - Target Application Architecture, Approved
 - Views corresponding to the selected viewpoints, addressing key stakeholder concerns
- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including such Application Architecture requirements as:
 - Gap analysis results
 - Applications interoperability requirements
 - Relevant technical requirements that will apply to this evolution of the architecture development cycle
 - Constraints on the Technology Architecture about to be designed
 - Updated business requirements, if appropriate
 - Updated data requirements, if appropriate
- Application Architecture components of an Architecture Roadmap (see the TOGAF Standard — Architecture Content)

The TOGAF Standard — Architecture Content contains a detailed description of architectural artifacts which might be produced in this phase.

7.5 Approach

7.5.1 Architecture Repository

As part of this phase, the architecture team will need to consider what relevant Application Architecture resources are available in the Architecture Repository (see the TOGAF Standard — Architecture Content).

In particular:

- Generic business models and related application models relevant to the organization's industry sector
- Application models relevant to common high-level business functions, such as electronic commerce, supply chain management, etc.

The Open Group has a Reference Model for Integrated Information Infrastructure (III-RM) — see the TOGAF® Series Guide: The TOGAF Integrated Information Infrastructure Reference Model (III-RM) — that focuses on the application-level components and services necessary to provide an integrated information infrastructure.

Chapter 8: Phase D: Technology Architecture

This chapter describes the development of a Technology Architecture for an architecture project.

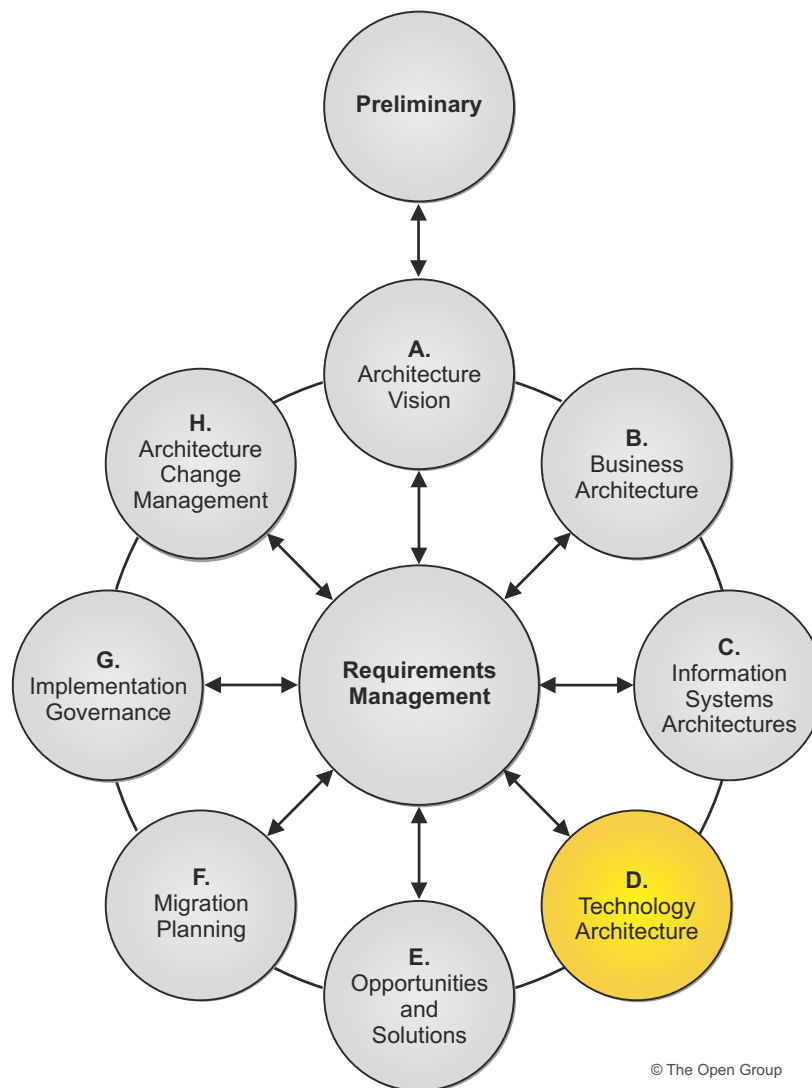


Figure 8-1 Phase D: Technology Architecture

8.1 Objectives

The objectives of Phase D are to:

- Develop the Target Technology Architecture that enables the Architecture Vision, target business, data, and application building blocks to be delivered through technology components and technology services, in a way that addresses the Statement of Architecture Work and stakeholder concerns
- Identify candidate Architecture Roadmap components based upon gaps between the Baseline and Target Technology Architectures

8.2 Inputs

This section defines the inputs to Phase D.

8.2.1 Reference Materials External to the Enterprise

- Architecture reference materials (see the TOGAF Standard — Architecture Content)
- Product information on candidate products

8.2.2 Non-Architectural Inputs

- Request for Architecture Work (see the TOGAF Standard — Architecture Content)
- Capability Assessment (see the TOGAF Standard — Architecture Content)
- Communications Plan (see the TOGAF Standard — Architecture Content)

8.2.3 Architectural Inputs

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Budget requirements
 - Governance and support strategy
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method

- Tailored architecture content (deliverables and artifacts)
- Configured and deployed tools
- Technology principles (see the TOGAF Standard — ADM Techniques), if existing
- Statement of Architecture Work (see the TOGAF Standard — Architecture Content)
- Architecture Vision (see the TOGAF Standard — Architecture Content)
- Architecture Repository (see the TOGAF Standard — Architecture Content), including:
 - Re-usable building blocks
 - Publicly available reference models
 - Organization-specific reference models
 - Organization standards
- Draft Architecture Definition Document, which may include Baseline and/or Target Architectures of any architecture domain
- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including:
 - Gap analysis results (from Business, Data, and Application Architectures)
 - Relevant technical requirements from previous phases
- Business, Data, and Application Architecture components of an Architecture Roadmap (see the TOGAF Standard — Architecture Content)

8.3 Steps

The level of detail addressed in Phase D will depend on the scope and goals of the overall architecture effort.

New technology building blocks being introduced as part of this effort will need to be defined in detail during Phase D. Existing technology building blocks to be supported in the target environment may need to be redefined in Phase D to ensure interoperability and fit-for-purpose within this specific Technology Architecture.

The order of the steps in Phase D as well as the time at which they are formally started and completed should be adapted to the situation at hand in accordance with the established Architecture Governance. In particular, determine whether in this situation it is appropriate to conduct Baseline Description or Target Architecture development first, as described in the TOGAF Standard — Applying the ADM.

All activities that have been initiated in these steps should be closed during the Finalize the Technology Architecture step (see [Section 8.3.8](#)). The documentation generated from these steps must be formally published in the Create/Update the Architecture Definition Document step (see [Section 8.3.9](#)).

The steps in Phase D are as follows:

- Select reference models, viewpoints, and tools (see [Section 8.3.1](#))
- Develop Baseline Technology Architecture Description (see [Section 8.3.2](#))
- Develop Target Technology Architecture Description (see [Section 8.3.3](#))
- Perform gap analysis (see [Section 8.3.4](#))
- Define candidate roadmap components (see [Section 8.3.5](#))
- Resolve impacts across the Architecture Landscape (see [Section 8.3.6](#))
- Conduct formal stakeholder review (see [Section 8.3.7](#))
- Finalize the Technology Architecture (see [Section 8.3.8](#))
- Create/Update the Architecture Definition Document (see [Section 8.3.9](#))

8.3.1 Select Reference Models, Viewpoints, and Tools

Review and validate the set of technology principles. These will normally form part of an overarching set of Architecture Principles. Guidelines for developing and applying principles, and a sample set of technology principles, are given in the TOGAF Standard — ADM Techniques.

Select relevant Technology Architecture resources (reference models, patterns, etc.) from the Architecture Repository (see the TOGAF Standard — Architecture Content), on the basis of the business drivers, stakeholders, and their concerns.

Select relevant Technology Architecture viewpoints that will enable the architect to demonstrate how the stakeholder concerns are being addressed in the Technology Architecture.

Identify appropriate tools and techniques to be used for capture, modeling, and analysis, in association with the selected viewpoints. Depending on the degree of sophistication required, these may comprise simple documents and spreadsheets, or more sophisticated modeling tools and techniques.

8.3.1.1 Determine Overall Modeling Process

For each viewpoint, select the models needed to support the specific view required, using the selected tool or method. Ensure that all stakeholder concerns are covered. If they are not, create new models to address them, or augment existing models (see above).

The process to develop a Technology Architecture incorporates the following steps:

- Define a taxonomy of technology services and logical technology components (including standards)
- Identify relevant locations where technology is deployed
- Carry out a physical inventory of deployed technology and abstract up to fit into the taxonomy
- Look at application and business requirements for technology
- Is the technology in place fit-for-purpose to meet new requirements (i.e., does it meet functional and non-functional requirements)?

- Refine the taxonomy
- Product selection (including dependent products)
- Determine configuration of the selected technology
- Determine impact:
 - Sizing and costing
 - Capacity planning
 - Installation/governance/migration impacts

In the earlier phases of the ADM, certain decisions made around service granularity and service boundaries will have implications on the technology component and the technology service. The areas where the Technology Architecture may be impacted will include the following:

- **Performance:** the granularity of the service will impact on technology service requirements
Coarse-grained services contain several units of functionality with potentially varying non-functional requirements, so platform performance should be considered. In addition, coarse-grained services can sometimes contain more information than actually required by the requesting system.
- **Maintainability:** if service granularity is too coarse, then introducing changes to that service becomes difficult and impacts the maintenance of the service and the platform on which it is delivered
- **Location and Latency:** services might interact with each other over remote links and inter-service communication will have in-built latency
Drawing service boundaries and setting the service granularity should consider platform/location impact of these inter-service communications.
- **Availability:** service invocation is subject to network and/or service failure
High communication availability is an important consideration during service decomposition and defining service granularity.

Product selection processes may occur within the Technology Architecture phase where existing products are re-used, incremental capacity is being added, or product selection decisions are a constraint during project initiation.

Where product selection deviates from existing standards, involves significant effort, or has wide-ranging impact, this activity should be flagged as an opportunity and addressed through the Opportunities & Solutions phase.

8.3.1.2 Identify Required Catalogs of Technology Building Blocks

Catalogs are inventories of the core assets of the business. Catalogs are hierarchical in nature and capture a decomposition of a metamodel entity and also decompositions across related model entities (e.g., technology service → logical technology component → physical technology component).

Catalogs form the raw material for development of matrices and diagrams and also act as a key resource for managing the business and IT capability.

The Technology Architecture should create technology catalogs as follows:

- Based on existing technology catalogs and analysis of applications carried out in the Application Architecture phase, 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 by examining products available on the market that provide the functionality and meet the required standards
- Classify products against the selected taxonomy if appropriate, extending the model as necessary to fit the classification of technology products in use
- If technology standards are currently in place, apply these to the technology component catalog to gain a baseline view of compliance with technology standards

The TOGAF Standard — Architecture Content contains a detailed description of catalogs which should be considered for development within a Technology Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

8.3.1.3 Identify Required Matrices

Matrices show the core relationships between related model entities.

Matrices form the raw material for development of diagrams and also act as a key resource for impact assessment.

The TOGAF Standard — Architecture Content contains a detailed description of matrices which should be considered for development within a Technology Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

8.3.1.4 Identify Required Diagrams

Diagrams present the Technology Architecture information from a set of different perspectives (viewpoints) according to the requirements of the stakeholders.

This activity provides a link between platform requirements and hosting requirements, as a single application may need to be physically located in several environments to support local access, development lifecycles, and hosting requirements.

For major baseline applications or application platforms (where multiple applications are hosted on the same infrastructure stack), produce a stack diagram showing how hardware, operating system, software infrastructure, and packaged applications combine.

If appropriate, extend the Application Architecture diagrams of software distribution to show how applications map onto the technology platform.

For each environment, produce a logical diagram of hardware and software infrastructure showing the contents of the environment and logical communications between components. Where available, collect capacity information on the deployed infrastructure.

For each environment, produce a physical diagram of communications infrastructure, such as routers, switches, firewalls, and network links. Where available, collect capacity information on the communications infrastructure.

The TOGAF Standard — Architecture Content contains a detailed description of diagrams which

should be considered for development within a Technology Architecture, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

8.3.1.5 *Identify Types of Requirement to be Collected*

Once the Technology Architecture catalogs, matrices, and diagrams have been developed, architecture modeling is completed by formalizing the technology-focused requirements for implementing the Target Architecture.

These requirements may:

- Relate to the technology domain
- Provide detailed guidance to be reflected during design and implementation to ensure that the solution addresses the original architecture requirements

Within this step, the architect should identify requirements that should be met by the architecture (see [Section 13.5.2](#)).

8.3.1.6 *Select Services*

The services portfolios are combinations of basic services from the service categories in the defined taxonomy that do not conflict. The combination of services are again tested to ensure support for the applications. This is a prerequisite to the later step of defining the architecture fully.

The previously identified requirements can provide more detailed information about:

- Requirements for organization-specific elements or pre-existing decisions (as applicable)
- Pre-existing and unchanging organizational elements (as applicable)
- Inherited external environment constraints

Where requirements demand definition of specialized services that are not identified in the TOGAF Standard, consideration should be given to how these might be replaced if standardized services become available in the future.

For each building block, build up a service description portfolio as a set of non-conflicting services. The set of services must be tested to ensure that the functionality provided meets application requirements.

8.3.2 **Develop Baseline Technology Architecture Description**

Develop a Baseline Description of the existing Technology Architecture, to support the Target Technology Architecture. The scope and level of detail to be defined will depend on the extent to which existing technology components are likely to be carried over into the Target Technology Architecture, and on whether architectural descriptions exist, as described in [Section 8.5](#).

Identify the relevant Technology Architecture building blocks, drawing on any artifacts held in the Architecture Repository. If nothing exists within the Architecture Repository, define each application in line with the Technology Portfolio catalog (see the TOGAF Standard — Architecture Content).

Begin by converting the description of the existing environment into the terms of the organization's taxonomy of technology services and technology components (e.g., the TOGAF Technical Reference Model (TRM)). This will allow the team developing the architecture to gain experience and understanding of the taxonomy. The team may be able to take advantage of a previous architectural definition, but it is assumed that some adaptation may be required to match the architectural definition techniques described as part of this process. Another important task is to set down a list of key questions which can be used later in the development process to measure the effectiveness of the new architecture.

Where new architecture models need to be developed to satisfy stakeholder concerns, use the models identified within Step 1 as a guideline for creating new architecture content to describe the Baseline Architecture.

8.3.3 Develop Target Technology Architecture Description

Develop a Target Description for the Technology Architecture, to the extent necessary to support the Architecture Vision, Target Business Architecture, and Target Information Systems Architecture. The scope and level of detail to be defined will depend on the relevance of the technology elements to attaining the Target Architecture, and on whether architectural descriptions exist. To the extent possible, identify the relevant Technology Architecture building blocks, drawing on the Architecture Repository (see the TOGAF Standard — Architecture Content).

A key process in the creation of a broad architectural model of the target system is the conceptualization of building blocks. ABBs describe the functionality and how they may be implemented without the detail introduced by configuration or detailed design. The method of defining building blocks, along with some general guidelines for their use in creating an architectural model, is described in the TOGAF Standard — Architecture Content.

Where new architecture models need to be developed to satisfy stakeholder concerns, use the models identified within Step 1 as a guideline for creating new architecture content to describe the Target Architecture.

If appropriate, investigate different Target Architecture alternatives and discuss these with stakeholders using the Architecture Alternatives and Trade-offs technique (see the TOGAF Standard — ADM Techniques).

8.3.4 Perform Gap Analysis

Verify the architecture models for internal consistency and accuracy:

- Perform trade-off analysis to resolve conflicts (if any) among the different views
- Validate that the models support the principles, objectives, and constraints
- Note changes to the viewpoint represented in the selected models from the Architecture Repository, and document
- Test architecture models for completeness against requirements

Identify gaps between the baseline and target, using the gap analysis technique as described in the TOGAF Standard — ADM Techniques.

8.3.5 Define Candidate Roadmap Components

Following the creation of a Baseline Architecture, Target Architecture, and gap analysis, a Technology Roadmap is required to prioritize activities over the coming phases.

This initial Technology Architecture roadmap will be used as raw material to support more detailed definition of a consolidated, cross-discipline roadmap within the Opportunities & Solutions phase.

8.3.6 Resolve Impacts Across the Architecture Landscape

Once the Technology Architecture is finalized, it is necessary to understand any wider impacts or implications.

At this stage, other architecture artifacts in the Architecture Landscape should be examined to identify:

- Does this Technology Architecture create an impact on any pre-existing architectures?
- Have recent changes been made that impact the Technology Architecture?
- Are there any opportunities to leverage work from this Technology Architecture in other areas of the organization?
- Does this Technology Architecture impact other projects (including those planned as well as those currently in progress)?
- Will this Technology Architecture be impacted by other projects (including those planned as well as those currently in progress)?

8.3.7 Conduct Formal Stakeholder Review

Check the original motivation for the architecture project and the Statement of Architecture Work against the proposed Technology Architecture, asking if it is fit for the purpose of supporting subsequent work in the other architecture domains. Refine the proposed Technology Architecture only if necessary.

8.3.8 Finalize the Technology Architecture

- Select standards for each of the building blocks, re-using as much as possible from the reference models selected from the Architecture Repository
- Fully document each building block
- Conduct a final cross-check of overall architecture against business goals; document the rationale for building block decisions in the architecture document
- Document the final requirements traceability report
- Document the final mapping of the architecture within the Architecture Repository; from the selected building blocks, identify those that might be re-used (working practices, roles, business relationships, job descriptions, etc.), and publish via the Architecture Repository

- Finalize all the work products, such as gap analysis

8.3.9 Create/Update the Architecture Definition Document

Document the rationale for building block decisions in the Architecture Definition Document.

Prepare the technology sections of the Architecture Definition Document, comprising some or all of:

- Fundamental functionality and attributes — semantic, unambiguous including security capability and manageability
- Dependent building blocks with required functionality and named interfaces
- Interfaces — chosen set, supplied (APIs, data formats, protocols, hardware interfaces, standards)
- Map to business/organizational entities and policies

If appropriate, use reports and/or graphics generated by modeling tools to demonstrate key views of the architecture. Route the document for review by relevant stakeholders, and incorporate feedback.

8.4 Outputs

The outputs of Phase D may include, but are not restricted to:

- Refined and updated versions of the Architecture Vision phase deliverables, where applicable:
 - Statement of Architecture Work (see the TOGAF Standard — Architecture Content), updated if necessary
 - Validated technology principles, or new technology principles (if generated here)
- Draft Architecture Definition Document (see the TOGAF Standard — Architecture Content), including:
 - Baseline Technology Architecture, Approved, if appropriate
 - Target Technology Architecture, Approved, including:
 - Technology Components and their relationships to information systems
 - Technology platforms and their decomposition, showing the combinations of technology required to realize a particular technology "stack"
 - Environments and locations — a grouping of the required technology into computing environments (e.g., development, production)
 - Expected processing load and distribution of load across technology components
 - Physical (network) communications
 - Hardware and network specifications

- Views corresponding to the selected viewpoints addressing key stakeholder concerns
- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including such Technology Architecture requirements as:
 - Gap analysis results
 - Requirements output from Phases B and C
 - Updated technology requirements
- Technology Architecture components of an Architecture Roadmap (see the TOGAF Standard — Architecture Content)

The TOGAF Standard — Architecture Content contains a detailed description of architectural artifacts which might be produced in this phase.

8.5 Approach

8.5.1 Emerging Technologies

The evolution of new technologies is a major driver for change in enterprises looking for new innovative ways of operating and improving their business. The Technology Architecture needs to capture the transformation opportunities available to the enterprise through the adoption of new technology.

While the Enterprise Architecture is led by business concerns, drivers for change are often found within evolving technology capabilities. As more digital innovations reach the market, stakeholders need to both anticipate and be open to technology-driven change. Part of Digital Transformation has arisen due to the convergence of telecommunications and computer capabilities which have opened up new ways of implementing infrastructures.

Solution development methods are also evolving to challenge traditional development methods and are putting pressure on the shared services and common use benefits of the traditional Enterprise Architecture approach. Without a strong Enterprise Architecture approach, the rapid adoption of changing technologies will cause discontinuities across the enterprise.

The flexibility of the TOGAF ADM enables technology change to become a driver and strategic resource rather than a recipient of Change Requests. As a result, the Technology Architecture may both drive business capabilities and respond to information system requirements at the same time.

8.5.2 Architecture Repository

As part of Phase D, the architecture team will need to consider what relevant Technology Architecture resources are available in the Architecture Repository (see the TOGAF Standard — Architecture Content).

In particular:

- Existing IT services as documented in the IT repository or IT service catalog
- The adopted technical reference model, if applicable

- Generic technology models relevant to the organization's industry sector
- Technology models relevant to Common Systems Architectures
 - The Open Group has a Reference Model for Integrated Information Infrastructure (III-RM) — see the TOGAF® Series Guide: The TOGAF Integrated Information Infrastructure Reference Model (III-RM) — that focuses on the application-level components and underlying services necessary to provide an integrated information infrastructure

Chapter 9: Phase E: Opportunities & Solutions

This chapter describes the process of identifying delivery vehicles (projects, programs, or portfolios) that effectively deliver the Target Architecture identified in previous phases.

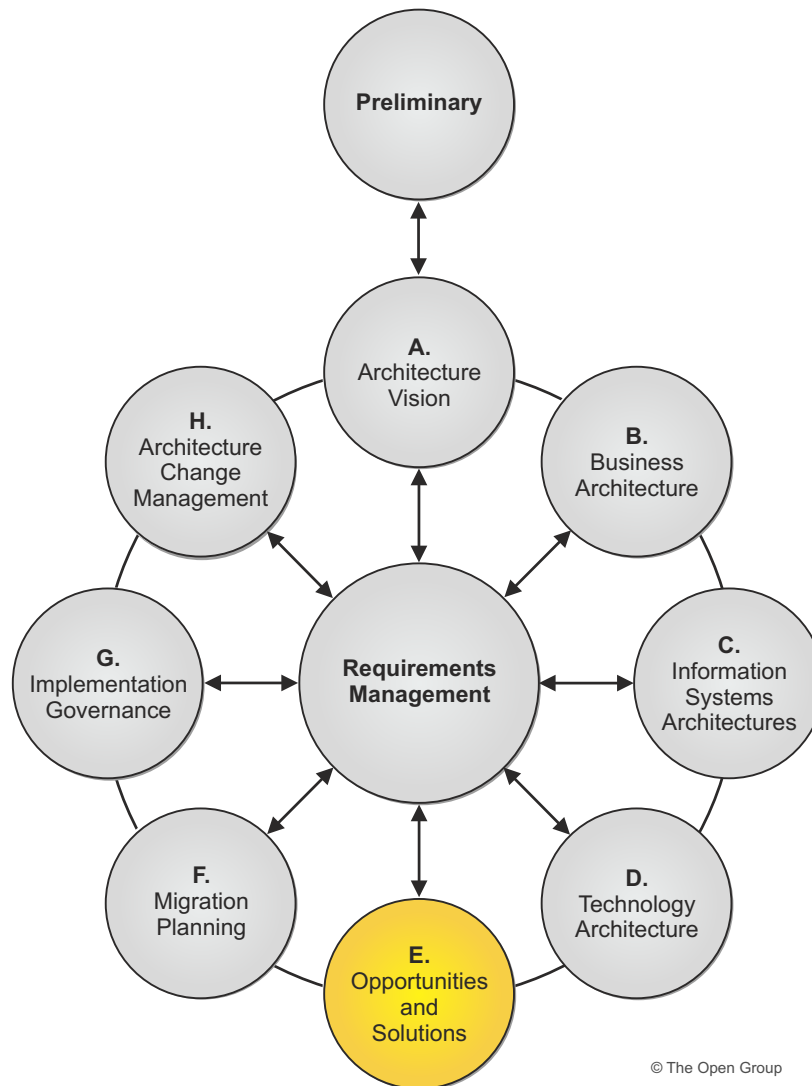


Figure 9-1 Phase E: Opportunities & Solutions

9.1 Objectives

The objectives of Phase E are to:

- Generate the initial complete version of the Architecture Roadmap, based upon the gap analysis and candidate Architecture Roadmap components from Phases B, C, and D
- Determine whether an incremental approach is required, and if so identify Transition Architectures that will deliver continuous business value
- Define the overall Solution Building Blocks (SBBs) to finalize the Target Architecture based on the ABBs

9.2 Inputs

This section defines the inputs to Phase E.

9.2.1 Reference Materials External to the Enterprise

- Architecture reference materials (see the TOGAF Standard — Architecture Content)
- Product information

9.2.2 Non-Architectural Inputs

- Request for Architecture Work (see the TOGAF Standard — Architecture Content)
- Capability Assessment (see the TOGAF Standard — Architecture Content)
- Communications Plan (see the TOGAF Standard — Architecture Content)
- Planning methodologies

9.2.3 Architectural Inputs

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Budget requirements
 - Governance and support strategy
- Governance models and frameworks for:
 - Corporate Business Planning

- Enterprise Architecture
- Portfolio, Program, Project Management
- System Development/Engineering
- Operations (Service)
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)
 - Configured and deployed tools
- Statement of Architecture Work (see the TOGAF Standard — Architecture Content)
- Architecture Vision (see the TOGAF Standard — Architecture Content)
- Architecture Repository (see the TOGAF Standard — Architecture Content), including:
 - Re-usable building blocks
 - Publicly available reference models
 - Organization-specific reference models
 - Organization standards
- Draft Architecture Definition Document, which may include Baseline and/or Target Architectures of any architecture domain
- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including:
 - Architectural requirements
 - Gap analysis results (from Business, Data, Application, and Technology Architecture)
 - IT Service Management requirements
- Change Requests for existing business programs and projects (see the TOGAF Standard — Architecture Content)
- Candidate Architecture Roadmap components from Phases B, C, and D

9.3 Steps

The level of detail addressed in Phase E will depend on the scope and goals of the overall architecture effort.

The order of the steps in Phase E as well as the time at which they are formally started and completed should be adapted to the situation at hand in accordance with the established Architecture Governance.

All activities that have been initiated in these steps must be closed during the Create the Architecture Roadmap & Implementation and Migration Plan step (see [Section 9.3.11](#)).

The steps in Phase E are as follows:

- Determine/confirm key corporate change attributes (see [Section 9.3.1](#))
- Determine business constraints for implementation (see [Section 9.3.2](#))
- Review and consolidate gap analysis results from Phases B to D (see [Section 9.3.3](#))
- Review consolidated requirements across related business functions (see [Section 9.3.4](#))
- Consolidate and reconcile interoperability requirements (see [Section 9.3.5](#))
- Refine and validate dependencies (see [Section 9.3.6](#))
- Confirm readiness and risk for business transformation (see [Section 9.3.7](#))
- Formulate Implementation and Migration Strategy (see [Section 9.3.8](#))
- Identify and group major work packages (see [Section 9.3.9](#))
- Identify Transition Architectures (see [Section 9.3.10](#))
- Create the Architecture Roadmap & Implementation and Migration Plan (see [Section 9.3.11](#))

9.3.1 Determine/Confirm Key Corporate Change Attributes

This step determines how the Enterprise Architecture can be best implemented to take advantage of the organization's business culture. This should include the creation of an Implementation Factor catalog (see the TOGAF Standard — ADM Techniques) to serve as a repository for architecture implementation and migration decisions. The step also includes assessments of the transition capabilities of the organization units involved (including culture and abilities), and assessments of the enterprise (including culture and skill sets).

The resulting factors from the assessments should be documented in the Implementation Factor catalog. For organizations where Enterprise Architecture is well established, this step can be simple, but the matrix has to be established so that it can be used as an archive and record of decisions taken.

9.3.2 Determine Business Constraints for Implementation

Identify any business drivers that would constrain the sequence of implementation. This should include a review of the business and strategic plans, at both a corporate and line-of-business level, and a review of the Enterprise Architecture Maturity Assessment.

9.3.3 Review and Consolidate Gap Analysis Results from Phases B to D

Consolidate and integrate the gap analysis results from the Business, Information Systems, and Technology Architectures (created in Phases B to D) and assess their implications with respect to potential solutions and inter-dependencies. This should be done by creating a Consolidated Gaps, Solutions, and Dependencies matrix, as shown in the TOGAF Standard — ADM Techniques, which will enable the identification of SBBs that could potentially address one or more gaps and their associated ABBs.

The SBBs are created based on the candidate roadmap components from Phases B, C, and D.

If appropriate, investigate different Target Architecture alternatives and discuss these with stakeholders using the Architecture Alternatives and Trade-offs technique (see the TOGAF Standard — ADM Techniques).

Review the Phase B, C, and D gap analysis results and consolidate them in a single list. The gaps should be consolidated along with potential solutions to the gaps and dependencies. A recommended technique for determining the dependencies is to use sets of views such as the Business Interaction matrix, the Data Entity/Business Function matrix, and the Application/Function matrix to completely relate elements from different architecture domains.

Rationalize the Consolidated Gaps, Solutions, and Dependencies matrix. Once all of the gaps have been documented, re-organize the gap list and place similar items together. When grouping the gaps, refer to the Implementation Factor catalog and review the implementation factors. Any additional factors should be added to the Implementation Factor catalog.

9.3.4 Review Consolidated Requirements Across Related Business Functions

Assess the requirements, gaps, solutions, and factors to identify a minimal set of requirements whose integration into work packages would lead to a more efficient and effective implementation of the Target Architecture across the business functions that are participating in the architecture. This functional perspective leads to the satisfaction of multiple requirements through the provision of shared solutions and services. The implications of this consolidation of requirements with respect to architectural components can be significant with respect to the provision of resources. For example, several requirements raised by several lines of business can be resolved through the provision of a shared set of Business Services and Application Services within a work package or project.

9.3.5 Consolidate and Reconcile Interoperability Requirements

Consolidate the interoperability requirements identified in previous phases. The Architecture Vision and Target Architectures, as well as the Implementation Factor catalog and Consolidated Gaps, Solutions, and Dependencies matrix, should be consolidated and reviewed to identify any constraints on interoperability required by the potential set of solutions.

A key outcome is to minimize interoperability conflicts, or to ensure such conflicts are addressed in the architecture. Re-used SBBs, Commercial Off-The-Shelf (COTS) products, and third-party service providers typically impose interoperability requirements that conflict. Any such conflicts must be addressed in the architecture, and conflicts must be considered across all architecture domains (Business, Data, Application, Technology).

There are two basic approaches to interoperability conflicts; either create a building block that transforms or translates between conflicting building blocks, or make a change to the specification of the conflicting building blocks.

9.3.6 Refine and Validate Dependencies

Refine the initial dependencies, ensuring that any constraints on the Implementation and Migration Plans are identified. There are several key dependencies that should be taken into account, such as dependencies on existing implementations of Business Services and Application Services or changes to them. Dependencies should be used for determining the sequence of implementation and identifying the co-ordination required. A study of the dependencies should group activities together, creating a basis for projects to be established. Examine the relevant projects and see whether logical increments of deliverables can be identified. The dependencies will also help to identify when the identified increments can be delivered. Once finished, an assessment of these dependencies should be documented as part of the Architecture Roadmap and any necessary Transition Architectures.

Addressing dependencies serves as the basis for most migration planning.

9.3.7 Confirm Readiness and Risk for Business Transformation

Review the findings of the Business Transformation Readiness Assessment previously conducted in Phase A and determine their impact on the Architecture Roadmap and the Implementation and Migration Strategy. It is important to identify, classify, and mitigate risks associated with the transformation effort. Risks should be documented in the Consolidated Gaps, Solutions, and Dependencies matrix.

9.3.8 Formulate Implementation and Migration Strategy

Create an overall Implementation and Migration Strategy that will guide the implementation of the Target Architecture, and structure any Transition Architectures. The first activity is to determine an overall strategic approach to implementing the solutions and/or exploiting opportunities. There are three basic approaches as follows:

- Greenfield: a completely new implementation
- Revolutionary: a radical change (i.e., switch on, switch off)
- Evolutionary: a strategy of convergence, such as parallel running or a phased approach to introduce new capabilities

Next, determine an approach for the overall strategic direction that will address and mitigate the risks identified in the Consolidated Gaps, Solutions, and Dependencies matrix. The most common implementation methodologies are:

- Quick win (snapshots)
- Achievable targets
- Value chain method

These approaches and the identified dependencies should become the basis for the creation of the work packages. This activity terminates with agreement on the Implementation and Migration Strategy for the enterprise.

9.3.9 Identify and Group Major Work Packages

Key stakeholders, planners, and the Enterprise Architects should assess the missing business capabilities identified in the Architecture Vision and Target Architecture.

Using the Consolidated Gaps, Solutions, and Dependencies matrix together with the Implementation Factor catalog, logically group the various activities into work packages.

Fill in the "Solution" column in the Consolidated Gaps, Solutions, and Dependencies matrix to recommend the proposed solution mechanisms. Indicate for every gap/activity whether the solution should be oriented towards a new development, or be based on an existing product, and/or use a solution that can be purchased. An existing system may resolve the requirement with minor enhancements. For new development this is a good time to determine whether the work should be conducted in-house or through a contract.

Classify every current system that is under consideration as:

- Mainstream: part of the future information system
- Contain: expected to be replaced or modified in the planning horizon (next three years)
- Replace: to be replaced in the planning horizon

Supporting top-level work packages should then in turn be decomposed into increments to deliver the capability increments. Analyze and refine these work packages or increments with respect to their business transformation issues and the strategic implementation approach. Finally, group the work packages into portfolios and projects within a portfolio, taking into consideration the dependencies and the strategic implementation approach.

9.3.10 Identify Transition Architectures

Where the scope of change to implement the Target Architecture requires an incremental approach, then one or more Transition Architectures may be necessary. These provide an ability to identify clear targets along the roadmap to realizing the Target Architecture. The Transition Architectures should provide measurable business value. The time-span between successive Transition Architectures does not have to be of uniform duration.

Development of Transition Architectures must be based upon the preferred implementation approach, the Consolidated Gaps, Solutions, and Dependencies matrix, the listing of projects and portfolios, as well as the enterprise's capacity for creating and absorbing change.

Determine where the difficult activities are, and unless there are compelling reasons, implement them after other activities that most easily deliver missing capability.

9.3.11 Create the Architecture Roadmap & Implementation and Migration Plan

Consolidate the work packages and Transition Architectures into the Architecture Roadmap, Draft, which describes a timeline of the progression from the Baseline Architecture to the Target Architecture. The timeline informs the Implementation and Migration Plan. The Architecture Roadmap frames the migration planning in Phase F. Identified Transition Architectures and work packages should have a clear set of outcomes. The Architecture Roadmap must demonstrate how the selection and timeline of Transition Architectures and work packages realizes the Target Architecture.

The detail of the Architecture Roadmap, Draft, should be expressed at a similar level of detail to the Architecture Definition Document developed in Phases B, C, and D. Where significant additional detail is required before implementation the architecture is likely transitioning to a different level. See the TOGAF Standard — Applying the ADM for techniques to manage iteration and different levels of detail.

The Implementation and Migration Plan must demonstrate the activity necessary to realize the Architecture Roadmap. The Implementation and Migration Plan forms the basis of the migration planning in Phase F. The detail of the Implementation and Migration Plan, Draft, must be aligned to the detail of the Architecture Roadmap and be sufficient to identify the necessary projects and resource requirements to realize the roadmap.

When creating the Implementation and Migration Plan there are many approaches to consider, such as a data-driven sequence, where application systems that create data are implemented first, then applications that process the data. A clear understanding of the dependencies and lifecycle of in-place SBBs is required for an effective Implementation and Migration Plan.

Finally, update the Architecture Vision, Architecture Definition Document, and Architecture Requirements Specification with any additional relevant outcomes from this phase.

9.4 Outputs

The outputs of Phase E may include, but are not restricted to:

- Refined and updated version of the Architecture Vision phase deliverables, where applicable, including:
 - Architecture Vision, including definition of types and degrees of interoperability
 - Statement of Architecture Work (see the TOGAF Standard — Architecture Content), updated if necessary
- Draft Architecture Definition Document (see the TOGAF Standard — Architecture Content), including:
 - Baseline Business Architecture, Approved, updated if necessary
 - Target Business Architecture, Approved, updated if necessary
 - Baseline Data Architecture, Approved, updated if necessary
 - Target Data Architecture, Approved, updated if necessary
 - Baseline Application Architecture, Approved, updated if necessary
 - Target Application Architecture, Approved, updated if necessary
 - Baseline Technology Architecture, Approved, updated if necessary
 - Target Technology Architecture, Approved, updated if necessary
 - Transition Architecture, number and scope as necessary
 - Views corresponding to the selected viewpoints addressing key stakeholder concerns

- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including:
 - Consolidated Gaps, Solutions, and Dependencies Assessment
- Capability Assessments, including:
 - Business Capability Assessment
 - IT Capability Assessment
- Architecture Roadmap (see the TOGAF Standard — Architecture Content), including:
 - Work package portfolio:
 - Work package description (name, description, objectives)
 - Functional requirements
 - Dependencies
 - Relationship to opportunity
 - Relationship to Architecture Definition Document and Architecture Requirements Specification
 - Relationship to any capability increments
 - Business value
 - Implementation Factor catalog
 - Impact
 - Identification of Transition Architectures, if any, including:
 - Relationship to Architecture Definition Document
 - Implementation recommendations:
 - Criteria measures of effectiveness
 - Risks and issues
 - Solution Building Blocks (SBBs)
- Implementation and Migration Plan, Draft, including:
 - Implementation and Migration Strategy

The TOGAF Standard — Architecture Content contains a detailed description of architectural artifacts which might be produced in this phase, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

9.5 Approach

Phase E concentrates on how to deliver the architecture. It takes into account the complete set of gaps between the Target and Baseline Architectures in all architecture domains, and logically groups changes into work packages within the enterprise's portfolios. This is an effort to build a best-fit roadmap that is based upon the stakeholder requirements, the enterprise's business transformation readiness, identified opportunities and solutions, and identified implementation constraints. The key is to focus on the final target while realizing incremental business value.

Phase E is the initial step in the creation of the Implementation and Migration Plan which is completed in Phase F. It provides the basis of a well considered Implementation and Migration Plan that is integrated into the enterprise's portfolio in Phase F.

The following four concepts are key to transitioning from developing to delivering a Target Architecture:

- Architecture Roadmap
- Work Packages
- Transition Architectures
- Implementation and Migration Plan

The Architecture Roadmap lists individual work packages in a timeline that will realize the Target Architecture.

Each work package identifies a logical group of changes necessary to realize the Target Architecture.

A Transition Architecture describes the enterprise at an architecturally significant state between the Baseline and Target Architectures. Transition Architectures provide interim Target Architectures upon which the organization can converge.

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

Chapter 10: Phase F: Migration Planning

This chapter addresses migration planning; that is, how to move from the Baseline to the Target Architectures by finalizing a detailed Implementation and Migration Plan.

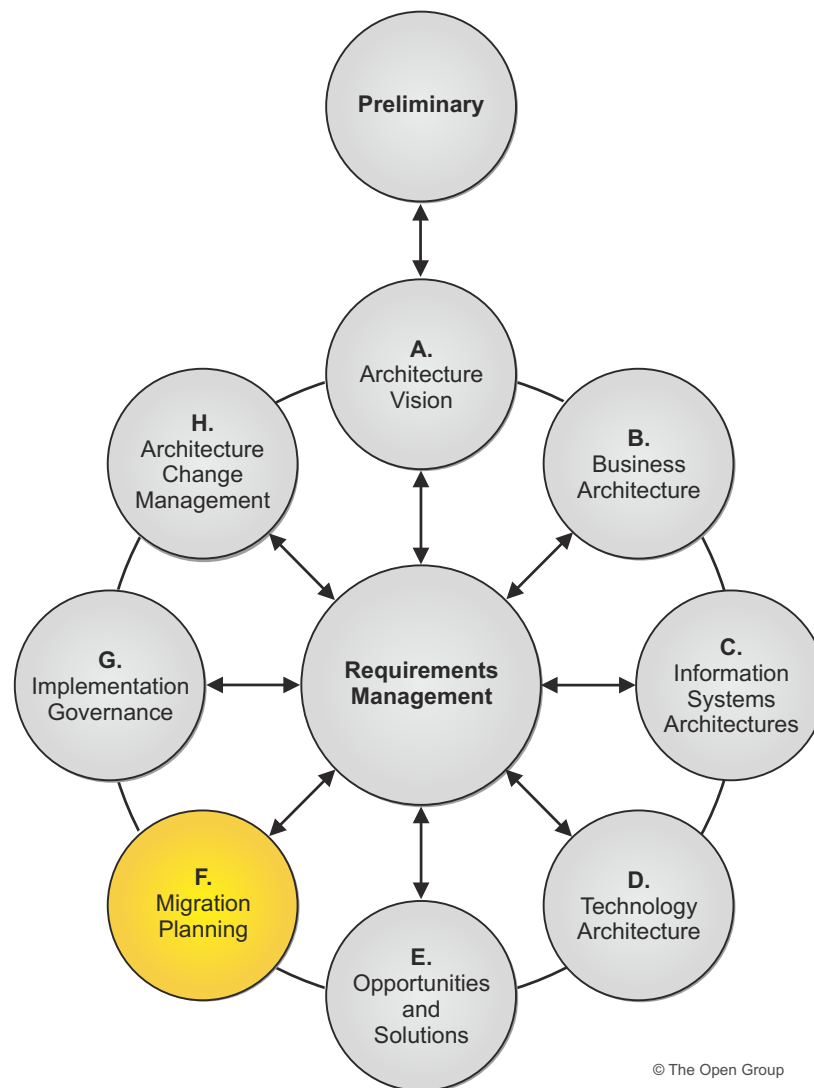


Figure 10-1 Phase F: Migration Planning

10.1 Objectives

The objectives of Phase F are to:

- Finalize the Architecture Roadmap and the supporting Implementation and Migration Plan
- Ensure that the Implementation and Migration Plan is co-ordinated with the enterprise's approach to managing and implementing change in the enterprise's overall change portfolio
- Ensure that the business value and cost of work packages and Transition Architectures is understood by key stakeholders

10.2 Inputs

This section defines the inputs to Phase F.

10.2.1 Reference Materials External to the Enterprise

- Architecture reference materials (see the TOGAF Standard — Architecture Content)

10.2.2 Non-Architectural Inputs

- Request for Architecture Work (see the TOGAF Standard — Architecture Content)
- Capability Assessment (see the TOGAF Standard — Architecture Content)
- Communications Plan (see TOGAF Standard — Architecture Content)

10.2.3 Architectural Inputs

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Budget requirements
 - Governance and support strategy
- Governance models and frameworks for:
 - Corporate Business Planning
 - Enterprise Architecture
 - Portfolio, Program, Project Management

- System Development/Engineering
 - Operations (Service)
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)
 - Configured and deployed tools
- Statement of Architecture Work (see the TOGAF Standard — Architecture Content)
- Architecture Vision (see the TOGAF Standard — Architecture Content)
- Architecture Repository (see the TOGAF Standard — Architecture Content), including:
 - Re-usable building blocks
 - Publicly available reference models
 - Organization-specific reference models
 - Organization standards
- Draft Architecture Definition Document, which may include Baseline and/or Target Architectures of any architecture domain, and/or Transition Architectures
- Draft Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including:
 - Architectural requirements
 - Gap analysis results (from Business, Data, Application, and Technology Architecture)
 - IT Service Management requirements
- Change Requests for existing business programs and projects (see the TOGAF Standard — Architecture Content)
- Architecture Roadmap, Draft (see the TOGAF Standard — Architecture Content), including:
 - Identification of work packages
 - Identification of Transition Architectures
 - Implementation Factor catalog
- Capability Assessment (see the TOGAF Standard — Architecture Content), including:
 - Business Capability Assessment
 - IT Capability Assessment
- Implementation and Migration Plan, Draft (see the TOGAF Standard — Architecture Content), including the high-level Implementation and Migration Strategy

10.3 Steps

The level of detail addressed in Phase F will depend on the scope and goals of the overall architecture effort.

The order of the steps in Phase F as well as the time at which they are formally started and completed should be adapted to the situation at hand in accordance with the established Architecture Governance.

All activities that have been initiated in these steps must be closed during the "Complete the Architecture Development Cycle and Document Lessons Learned" step (see [Section 10.3.7](#)).

The steps in Phase F are as follows:

- Confirm management framework interactions for Implementation and Migration Plan (see [Section 10.3.1](#))
- Assign a business value to each work package (see [Section 10.3.2](#))
- Estimate resource requirements, project timings, and availability/delivery vehicle (see [Section 10.3.3](#))
- Prioritize the migration projects through the conduct of a cost/benefit assessment and risk validation (see [Section 10.3.4](#))
- Confirm Architecture Roadmap and update Architecture Definition Document (see [Section 10.3.5](#))
- Complete the Implementation and Migration Plan (see [Section 10.3.6](#))
- Complete the architecture development cycle and document lessons learned (see [Section 10.3.7](#))

10.3.1 Confirm Management Framework Interactions for the Implementation and Migration Plan

This step is about co-ordinating the Implementation and Migration Plan with the management frameworks within the organization. There are typically four management frameworks that have to work closely together for the Implementation and Migration Plan to succeed:

- **Business Planning** that conceives, directs, and provides the resources for all of the activities required to achieve concrete business objectives/outcomes
- **Enterprise Architecture** that structures and gives context to all enterprise activities delivering concrete business outcomes primarily but not exclusively in the IT domain
- **Project/Portfolio Management** that co-ordinates, designs, and builds the business systems that deliver the concrete business outcomes
- **Operations Management** that integrates, operates, and maintains the deliverables that deliver the concrete business outcomes

The Implementation and Migration Plan will impact the outputs of each of these frameworks and consequently has to be reflected in them. In the course of this step, understand the frameworks within the organization and ensure that these plans are co-ordinated and inserted (in a summary format) within the plans of each one of these frameworks.

The outcome of this step may well be that the Implementation and Migration Plan could be part

of a different plan produced by another one of the frameworks with Enterprise Architecture participation.

10.3.2 Assign a Business Value to Each Work Package

Establish and assign a business value to each of the work packages. The intent is to first establish what constitutes business value within the organization, how value can be measured, and then apply this to each one of the projects and project increments.

There are several issues to address in this activity:

- **Performance Evaluation Criteria** are used by portfolio and capability managers to approve and monitor the progress of the architecture transformation
- **Return-on-Investment Criteria** have to be detailed and signed off by the various executive stakeholders
- **Business Value** has to be defined as well as techniques, such as the value chain, which are to be used to illustrate the role in achieving tangible business outcomes

Business value will be used by portfolio and capability managers to allocate resources and, in cases where there are cutbacks, business value in conjunction with return on investment can be used to determine whether an endeavor proceeds, is delayed, or is canceled.

- **Critical Success Factors (CSFs)** should be established to define success for a project and/or project increment

These will provide managers and implementers with a gauge as to what constitutes a successful implementation.

- **Measures of Effectiveness (MOE)** are often performance criteria and many corporations include them in the CSFs

Where they are treated discretely, it should be clear as to how these criteria are to be grouped.

- **Strategic Fit** based upon the overall Enterprise Architecture (all tiers) will be the critical factor for allowing the approval of any new project or initiative and for determining the value of any deliverable

Use the work packages as a basis of identifying projects that will be in the Implementation and Migration Plan. The identified projects will be fully developed in other steps in Phase F. The projects, and project increments, may require adjustment of the Architecture Roadmap and Architecture Definition Document.

Risks should then be assigned to the projects and project increments by aggregating risks identified in the Consolidated Gaps, Solutions, and Dependencies Matrix (from Phase E).

Estimate the business value for each project using the Business Value Assessment Technique (see the TOGAF Standard — ADM Techniques).

10.3.3 Estimate Resource Requirements, Project Timings, and Availability/Delivery Vehicle

This step determines the required resources and times for each project and their increments and provides the initial cost estimates. The costs should be broken down into capital (to create the capability) and operations and maintenance (to run and sustain the capability). Opportunities should be identified where the costs associated with delivering new and/or better capability can be offset by decommissioning existing systems. Assign required resources to each activity and aggregate them at the project increment and project level.

10.3.4 Prioritize the Migration Projects through the Conduct of a Cost/Benefit Assessment and Risk Validation

Prioritize the projects by ascertaining their business value against the cost of delivering them. The approach is to first determine, as clearly as possible, the net benefit of all of the SBBs delivered by the projects, and then verify that the risks have been effectively mitigated and factored in. Afterwards, the intent is to gain the requisite consensus to create a prioritized list of projects that will provide the basis for resource allocation.

It is important to discover all costs, and to ensure that decision-makers understand the net benefit over time.

Review the risks to ensure that the risks for the project deliverables have been mitigated as much as possible. The project list is then updated with risk-related comments.

Have the stakeholders agree upon a prioritization of the projects. Prioritization criteria will use elements identified in the creation of the draft Architecture Roadmap in Phase E as well as those relating to individual stakeholder agendas. Notice that it is possible for a project to earn a high priority if it provides a critical deliverable on the path to some large benefit, even if the immediate benefit of the project itself is small.

Formally review the risk assessment and revise it as necessary ensuring that there is a full understanding of the residual risk associated with the prioritization and the projected funding line.

10.3.5 Confirm Architecture Roadmap and Update Architecture Definition Document

Update the Architecture Roadmap including any Transition Architectures. Review the work to date to assess what the time-spans between Transition Architecture should be, taking into consideration the increments in business value and capability and other factors, such as risk. Once the capability increments have been finalized, consolidate the deliverables by project. This will result in a revised Architecture Roadmap.

This is needed in order to co-ordinate the development of several concurrent instances of the various architectures. A Transition Architecture State Evolution Table (see the TOGAF Standard — ADM Techniques) can be used to show the proposed state of the domain architectures at various levels of detail.

If the implementation approach has shifted as a result of confirming the implementation increments, update the Architecture Definition Document. This may include assigning project objectives and aligning projects and their deliverables with the Transition Architectures to create an Architecture Definition Increments Table (see the TOGAF Standard — ADM Techniques).

10.3.6 Complete the Implementation and Migration Plan

Generate the completed Implementation and Migration Plan. Much of the detail for the plan has already been gathered and this step brings it all together using accepted planning and management techniques. This should include integrating all of the projects and activities as well as dependencies and impact of change into a project plan. Any Transition Architectures will act as portfolio milestones.

All external dependencies should be captured and included, and the overall availability of resources assessed. Project plans may be included within the Implementation and Migration Plan.

10.3.7 Complete the Architecture Development Cycle and Document Lessons Learned

This step transitions governance from the development of the architecture to the realization of the architecture. If the maturity of the Architecture Capability warrants, an Implementation Governance Model may be produced (see the TOGAF Standard — Architecture Content).

Lessons learned during the development of the architecture should be documented and captured by the appropriate governance process in Phase H as inputs to managing the Architecture Capability.

The detail of the Architecture Roadmap and the Implementation and Migration Plan should be expressed at a similar level of detail to the Architecture Definition Document developed in Phases B, C, and D. Where significant additional detail is required by the next phase the architecture is likely transitioning to a different level. Depending upon the level of the Target Architecture and Implementation and Migration Plan it may be necessary to iterate another ADM cycle at a lower level of detail. See the TOGAF Standard — Applying the ADM for techniques to manage iteration and different levels of detail.

10.4 Outputs

The outputs of Phase F may include, but are not restricted to:

- Implementation and Migration Plan, Approved (see the TOGAF Standard — Architecture Content), including:
 - Implementation and Migration Strategy
 - Project and portfolio breakdown of the implementation:
 - Allocation of work packages to project and portfolio
 - Capabilities delivered by projects
 - Relationship to Target Architecture and any Transition Architectures
 - Milestones and timing
 - Work breakdown structure
 - Project charters (optional):
 - Related work packages

- Business value
- Risk, issues, assumptions, dependencies
- Resource requirements and costs
- Benefits of migration
- Estimated costs of migration options
- Finalized Architecture Definition Document (see the TOGAF Standard — Architecture Content), including:
 - Finalized Transition Architectures, if any
- Finalized Architecture Requirements Specification (see the TOGAF Standard — Architecture Content)
- Finalized Architecture Roadmap (see the TOGAF Standard — Architecture Content)
- Re-Usable ABBs (see the TOGAF Standard — Architecture Content)
- Requests for Architecture Work (see the TOGAF Standard — Architecture Content) for a new iteration of the ADM cycle (if any)
- Implementation Governance Model (if any) (see the TOGAF Standard — Architecture Content)
- Change Requests for the Architecture Capability arising from lessons learned

10.5 Approach

The focus of Phase F is the creation of an Implementation and Migration Plan in co-operation with the project and portfolio managers.

Phase E provides an incomplete Architecture Roadmap and Implementation and Migration Plan that address the Statement of Architecture Work. In Phase F this Roadmap and the Implementation and Migration Plan are integrated with the enterprise's other change activity.

Activities include assessing the dependencies, costs, and benefits of the various migration projects within the context of the enterprise's other activity. The Draft Architecture Roadmap, and Draft Implementation and Migration Plan, from Phase E will form the basis of the final Implementation and Migration Plan that will include portfolio and project-level detail.

The architecture development cycle should then be completed and lessons learned documented to enable continuous process improvement.

Chapter 11: Phase G: Implementation Governance

This chapter provides an architectural oversight of the implementation.

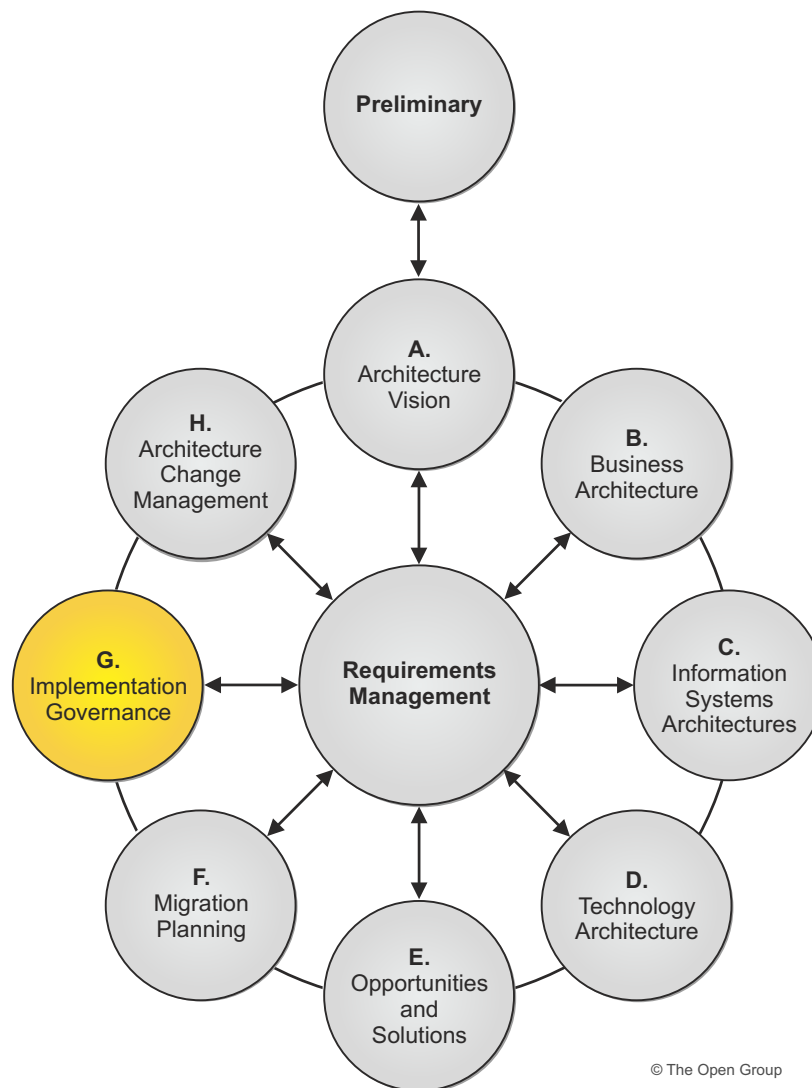


Figure 11-1 Phase G: Implementation Governance

11.1 Objectives

The objectives of Phase G are to:

- Ensure conformance with the Target Architecture by implementation projects
- Perform appropriate Architecture Governance functions for the solution and any implementation-driven architecture Change Requests

11.2 Inputs

This section defines the inputs to Phase G.

11.2.1 Reference Materials External to the Enterprise

- Architecture reference materials (see the TOGAF Standard — Architecture Content)

11.2.2 Non-Architectural Inputs

- Request for Architecture Work (see the TOGAF Standard — Architecture Content)
- Capability Assessment (see the TOGAF Standard — Architecture Content)

11.2.3 Architectural Inputs

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Budget requirements
 - Governance and support strategy
- Tailored Architecture Framework (see TOGAF Standard — Architecture Content), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)
 - Configured and deployed tools
- Statement of Architecture Work (see the TOGAF Standard — Architecture Content)
- Architecture Vision (see the TOGAF Standard — Architecture Content)

- Architecture Repository (see the TOGAF Standard — Architecture Content), including:
 - Re-usable building blocks
 - Publicly available reference models
 - Organization-specific reference models
 - Organization standards
- Architecture Definition Document (see the TOGAF Standard — Architecture Content)
- Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including:
 - Architectural requirements
 - Gap analysis results (from Business, Data, Application, and Technology Architectures)
- Architecture Roadmap (see the TOGAF Standard — Architecture Content)
- Architecture Governance Framework (see the TOGAF Standard — Enterprise Architecture Capability and Governance)
- Implementation Governance Model (see the TOGAF Standard — Architecture Content)
- Architecture Contract (standard) (see the TOGAF Standard — Enterprise Architecture Capability and Governance)
- Request for Architecture Work (see the TOGAF Standard — Architecture Content) identified during Phases E and F
- Implementation and Migration Plan (see the TOGAF Standard — Architecture Content)

11.3 Steps

The level of detail addressed in Phase G will depend on the scope and goals of the overall architecture effort.

The order of the steps in Phase G as well as the time at which they are formally started and completed should be adapted to the situation at hand in accordance with the established Architecture Governance.

The steps in Phase G are as follows:

- Confirm scope and priorities for deployment with development management (see [Section 11.3.1](#))
- Identify deployment resources and skills (see [Section 11.3.2](#))
- Guide development of solutions deployment (see [Section 11.3.3](#))
- Perform Enterprise Architecture Compliance reviews (see [Section 11.3.4](#))
- Implement business and IT operations (see [Section 11.3.5](#))
- Perform post-implementation review and close the implementation (see [Section 11.3.6](#))

11.3.1 Confirm Scope and Priorities for Deployment with Development Management

- Review migration planning outputs and produce recommendations on deployment
- Identify Enterprise Architecture priorities for development teams
- Identify deployment issues and make recommendations
- Identify building blocks for replacement, update, etc.
- Perform gap analysis on Enterprise Architecture and solutions framework

The gaps in the existing enterprise solutions framework need to be identified and the specific SBBs required to fill these gaps will be identified by the Solution Architects. These SBBs may have a one-to-one or many-to-one relationship with the projects. The Solution Architects need to define exactly how this will be done. There may be other projects working on these same capabilities and the Solution Architects need to ensure that they can leverage best value from these investments.

- Produce a gap analysis report

11.3.2 Identify Deployment Resources and Skills

The project resources will include the development resources which will need to be educated in the overall Enterprise Architecture deliverables and expectations from the specific development and implementation projects.

The following considerations should be addressed in this step:

- Identify system development methods required for solutions development
 - Note:** There are a range of systems development methods and tools available to the project teams. The method should ideally be able to interoperate with the architecture outputs; for example, generate code from architecture artifacts delivered to date. This could be achieved through the use of modeling languages used for the Enterprise Architecture development that may be captured as inputs to the systems development tools and thereby reduce the cost of solutions development.
- Ensure that the systems development method enables feedback to the architecture team on designs

11.3.3 Guide Development of Solutions Deployment

- Formulate project recommendation

For each separate implementation and deployment project, do the following:

- Document scope of individual project in impact analysis
- Document strategic requirements (from the architectural perspective) in impact analysis
- Document Change Requests (such as support for a standard interface) in impact analysis

- Document rules for conformance in impact analysis
 - Document timeline requirements from roadmap in impact analysis
- Document Architecture Contract
 - Obtain signature from all developing organizations and sponsoring organization
- Update Enterprise Continuum directory and repository for solutions
- Guide development of business & IT operating models for services
- Provide service requirements derived from Enterprise Architecture
- Guide definition of business & IT operational requirements
- Carry out gap analysis between the Solution Architecture and operations
- Produce Implementation Plan

11.3.4 Perform Enterprise Architecture Compliance Reviews

- Review ongoing implementation governance and Architecture Compliance for each building block
- Conduct post-development reviews
- Close development part of deployment projects

11.3.5 Implement Business and IT Operations

- Carry out the deployment projects including: IT services delivery implementation; business services delivery implementation; skills development & training implementation; communications documentation publication
- Publish new Baseline Architectures to the Architecture Repository and update other impacted repositories, such as operational configuration management stores

11.3.6 Perform Post-Implementation Review and Close the Implementation

- Conduct post-implementation reviews
- Publish reviews and close projects

Closure on Phase G will be when the solutions are fully deployed once.

11.4 Outputs

The outputs of Phase G may include, but are not restricted to:

- Architecture Contract (signed) (see the TOGAF Standard — Enterprise Architecture Capability and Governance), as recommended in the architecture-compliant implemented architectures
 - Compliance Assessments (see the TOGAF Standard — Architecture Content)
 - Change Requests (see the TOGAF Standard — Architecture Content)
 - Architecture-compliant solutions deployed including:
 - The architecture-compliant implemented system
- Note:** The implemented system is actually an output of the development process. However, given the importance of this output, it is stated here as an output of the ADM. The direct involvement of architecture staff in implementation will vary according to organizational policy, as described in the TOGAF Standard — Enterprise Architecture Capability and Governance.
- Populated Architecture Repository
 - Architecture compliance recommendations and dispensations
 - Recommendations on service delivery requirements
 - Recommendations on performance metrics
 - Service-Level Agreements (SLAs)
 - Architecture Vision, updated post-implementation
 - Architecture Definition Document, updated post-implementation
 - Business and IT operating models for the implemented solution
 - Architecture Building Blocks (ABBs)

11.5 Approach

It is here that all the information for successful management of the various implementation projects is brought together. Note that, in parallel with Phase G, there is the execution of an organizational-specific development process, where the actual development happens.

To enable early realization of business value and benefits, and to minimize the risk in the transformation and migration program, the favored approach is to deploy the Target Architecture as a series of transitions. Each transition represents an incremental step towards the target, and each delivers business benefit in its own right. Therefore, the overall approach in Phase G is to:

- Establish an implementation program that will enable the delivery of the Transition Architectures agreed for implementation during the Migration Planning phase
- Adopt a phased deployment schedule that reflects the business priorities embodied in the Architecture Roadmap
- Follow the organization's standard for corporate, IT, and Architecture Governance

- Use the organization's established portfolio/program management approach, where this exists
- Define an operations framework to ensure the effective long life of the deployed solution

Phase G establishes the connection between architecture and implementation organization, through the Architecture Contract.

Project details are developed, including:

- Name, description, and objectives
- Scope, deliverables, and constraints
- Measures of effectiveness
- Acceptance criteria
- Risks and issues

Implementation governance is closely allied to overall Architecture Governance, which is discussed in the TOGAF Standard — Enterprise Architecture Capability and Governance.

A key aspect of Phase G is ensuring compliance with the defined architecture(s), not only by the implementation projects, but also by other ongoing projects within the enterprise. The considerations involved with this are explained in detail in the TOGAF Standard — Enterprise Architecture Capability and Governance.

Chapter 12: Phase H: Architecture Change Management

This chapter looks at establishing procedures for managing change to the new architecture.

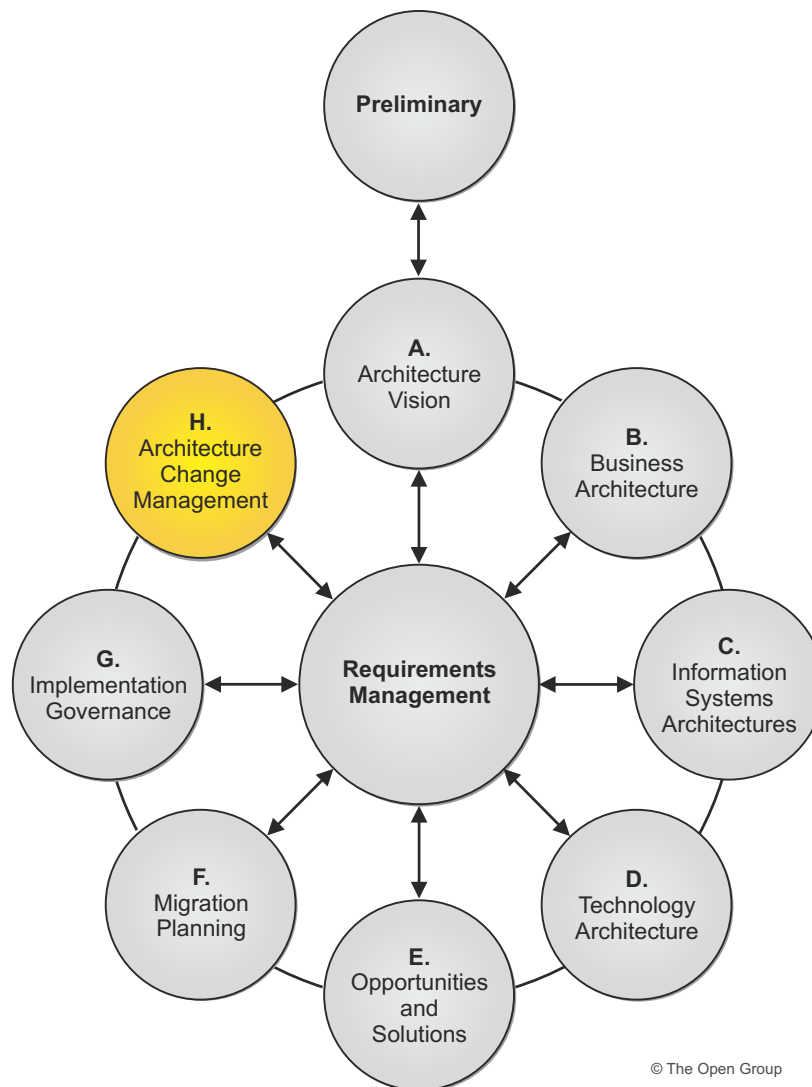


Figure 12-1 Phase H: Architecture Change Management

12.1 Objectives

The objectives of Phase H are to:

- Ensure that the architecture development cycle is maintained
- Ensure that the Architecture Governance Framework is executed
- Ensure that the Enterprise Architecture Capability meets current requirements

12.2 Inputs

This section defines the inputs to Phase H.

12.2.1 Reference Materials External to the Enterprise

- Architecture reference materials (see the TOGAF Standard — Architecture Content)

12.2.2 Non-Architectural Inputs

- Request for Architecture Work (see the TOGAF Standard — Architecture Content)

12.2.3 Architectural Inputs

- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Budget requirements
 - Governance and support strategy
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)
 - Configured and deployed tools
- Statement of Architecture Work (see the TOGAF Standard — Architecture Content)
- Architecture Vision (see the TOGAF Standard — Architecture Content)

- Architecture Repository (see the TOGAF Standard — Architecture Content), including:
 - Re-usable building blocks
 - Publicly available reference models
 - Organization-specific reference models
 - Organization standards
- Architecture Definition Document (see the TOGAF Standard — Architecture Content)
- Architecture Requirements Specification (see the TOGAF Standard — Architecture Content), including:
 - Gap analysis results (from Business, Data, Application, and Technology Architectures)
 - Architectural requirements
- Architecture Roadmap (see the TOGAF Standard — Architecture Content)
- Change Request (see the TOGAF Standard — Architecture Content) — technology changes:
 - New technology reports
 - Asset management cost reduction initiatives
 - Technology withdrawal reports
 - Standards initiatives
- Change Request (see the TOGAF Standard — Architecture Content) — business changes:
 - Business developments
 - Business exceptions
 - Business innovations
 - Business technology innovations
 - Strategic change developments
- Change Request (see the TOGAF Standard — Architecture Content) — from lessons learned
- Implementation Governance Model (see the TOGAF Standard — Architecture Content)
- Architecture Contract (signed) (see the TOGAF Standard — Enterprise Architecture Capability and Governance)
- Compliance Assessments (see the TOGAF Standard — Architecture Content)
- Implementation and Migration Plan (see the TOGAF Standard — Architecture Content)

12.3 Steps

The level of detail addressed in Phase H will depend on the scope and goals of the overall architecture effort.

The order of the steps in Phase H as well as the time at which they are formally started and completed should be adapted to the situation at hand in accordance with the established Architecture Governance.

The steps in Phase H are as follows:

- Establish value realization process (see [Section 12.3.1](#))
- Deploy monitoring tools (see [Section 12.3.2](#))
- Manage risks (see [Section 12.3.3](#))
- Provide analysis for architecture change management (see [Section 12.3.4](#))
- Develop change requirements to meet performance targets (see [Section 12.3.5](#))
- Manage governance process (see [Section 12.3.6](#))
- Activate the process to implement change (see [Section 12.3.7](#))

12.3.1 Establish Value Realization Process

Influence business projects to exploit the Enterprise Architecture for value realization (outcomes).

12.3.2 Deploy Monitoring Tools

Ensure monitoring tools are deployed and applied to enable the following:

- Monitor technology changes which could impact the Baseline Architecture
- Monitor business changes which could impact the Baseline Architecture
- Business value tracking; e.g., investment appraisal method to determine value metrics for the business objectives
- Monitor Enterprise Architecture Capability maturity
- Track and assess asset management programs
- Track the Quality of Service (QoS) performances and usage
- Determine and track business continuity requirements

12.3.3 Manage Risks

Manage Enterprise Architecture risks and provide recommendations for IT strategy.

12.3.4 Provide Analysis for Architecture Change Management

Provide analysis for architecture change management:

- Analyze performance
- Conduct Enterprise Architecture performance reviews with service management
- Assess Change Requests and reporting to ensure that the expected value realization and Service-Level Agreement (SLA) expectations of the customers are met
- Undertake a gap analysis of the performance of the Enterprise Architecture
- Ensure change management requests adhere to the Enterprise Architecture Governance and framework

12.3.5 Develop Change Requirements to Meet Performance Targets

Make recommendations on change requirements to meet performance targets and development of position to act.

12.3.6 Manage Governance Process

Manage governance process and framework for architecture:

- Arrange meeting of Architecture Board (or other Governing Council)
- Hold meeting of the Architecture Board with the aim of the meeting to decide on handling changes (technology and business and dispensations)

12.3.7 Activate the Process to Implement Change

Activate the architecture process to implement change:

- Produce a new Request for Architecture Work and request for investment
- Ensure any changes implemented in this phase are captured and documented in the Architecture Repository

12.4 Outputs

The outputs of Phase H may include, but are not restricted to:

- Architecture updates (for maintenance changes)
- Changes to architecture framework and principles (for maintenance changes)
- New Request for Architecture Work (see the TOGAF Standard — Architecture Content), to move to another cycle (for major changes)
- Statement of Architecture Work (see the TOGAF Standard — Architecture Content), updated if necessary
- Architecture Contract (see the TOGAF Standard — Enterprise Architecture Capability and Governance), updated if necessary
- Compliance Assessments (see the TOGAF Standard — Architecture Content), updated if necessary

12.5 Approach

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.

This process will typically provide for the continual monitoring of such things as governance requests, new developments in technology, and changes in the business environment. When changes are identified, change management will determine whether to formally initiate a new architecture evolution cycle.

Additionally, the architecture change management process aims to establish and support the implemented Enterprise Architecture as a dynamic architecture; that is, one having the flexibility to evolve rapidly in response to changes in the technology and business environment.

Monitoring business growth and decline is a critical aspect of this phase. Usage of the Enterprise Architecture is the most important part of the architecture development cycle. All too often the business has been left with an Enterprise Architecture that works for the organization of yesterday but may not give back sufficient capability to meet the needs of the enterprise of today and tomorrow.

In many cases the architecture continues to fit, but the solutions underlying them may not, and some changes are required. The Enterprise Architect needs to be aware of these change requirements and considers this an essential part of constant renewal of the architecture.

Capacity measurement and recommendations for planning are a key aspect of this phase. While the architecture has been built to deliver a steady state Business Architecture with agreed capacity during the lifecycle of this Enterprise Architecture, the growth or decline in usage needs to be continually assessed to ensure that maximum business value is achieved.

For example, some Solution Architectures may not lend themselves to be scalable by a large factor — say 10 — or alternative solutions may be more economic when scaled up. While the architecture specifications may not change, the solutions or their operational context may change.

If the performance management and reporting has been built into the work products through

previous phases, then this phase is about ensuring the effectiveness of these. If there needs to be additional monitoring or reporting, then this phase will handle the changes.

The value and change management process, once established, will determine:

- The circumstances under which the Enterprise Architecture, or parts of it, will be permitted to change after deployment, and the process by which that will happen
- The circumstances under which the architecture development cycle will be initiated again to develop a new architecture

The architecture change management process is very closely related to the Architecture Governance processes of the enterprise, and to the management of the Architecture Contract (see the TOGAF Standard — Enterprise Architecture Capability and Governance) between the architecture function and the business users of the enterprise.

In Phase H it is critical that the governance body establish criteria to judge whether a Change Request warrants just an architecture update or whether it warrants starting a new cycle of the ADM. It is especially important to avoid "creeping elegance", and the governance body must continue to look for changes that relate directly to business value.

An Architecture Compliance report should state whether the change is compliant to the current architecture. If it is non-compliant, an exemption may be granted with valid rationale. If the change has high impact on the architecture, then a strategy to manage its impact should be defined.

Guidelines for establishing these criteria are difficult to prescribe, as many companies accept risk differently, but as the ADM is exercised, the maturity level of the governance body will improve, and criteria will become clear for specific needs.

12.5.1 Drivers for Change

The main purpose for the development of the Enterprise Architecture so far has been strategic direction and top-down architecture and project generation to achieve corporate capabilities. However, Enterprise Architecture does not operate in a vacuum. There is usually an existing infrastructure and business which is already providing value.

There are also probably drivers for change which are often bottom-up, based upon modifying the existing infrastructure to enhance functionality. Enterprise Architecture changes this paradigm by a strategic top-down approach to a degree, although the delivery of increments makes the equation more complex.

There are three ways to change the existing infrastructure that have to be integrated:

- Strategic, top-down directed change to enhance or create new capability (capital)
- Bottom-up changes to correct or enhance capability (operations and maintenance) for infrastructure under operations management
- Experiences with the previously delivered project increments in the care of operations management, but still being delivered by ongoing projects

Governance will have to handle the co-ordination of these Requests for Change, plus there needs to be a lessons learned process to allow for problems with the recently delivered increments to be resolved and changes made to the Target Architectures being designed and planned.

A lessons learned process ensures that mistakes are made once and not repeated. They can come from anywhere and anyone and cover any aspect of the Enterprise Architecture at any level (strategic, Enterprise Architecture definition, transition, or project). Often an Enterprise Architecture-related lesson may be an indirect outcome of a lesson learned elsewhere in the organization.

The Architecture Board (see the TOGAF Standard — Enterprise Architecture Capability and Governance) assesses and approves Requests for Change (RFC). An RFC is typically in response to known problems but can also include improvements. A challenge for the Architecture Board when handling an RFC is to determine whether it should be approved or whether a project in a Transition Architecture will resolve the issue.

When assessing project or solution fit into the architecture, there may also be the case when an innovative solution or RFC drives a change in the architecture.

In addition, there are many technology-related drivers for architecture Change Requests. For example:

- New technology reports
- Asset management cost reductions
- Technology withdrawal
- Standards initiatives

This type of Change Request is normally manageable primarily through an enterprise's change management and Architecture Governance processes.

In addition, there are business drivers for architecture change, including:

- Business-as-usual developments
- Business exceptions
- Business innovations
- Business technology innovations
- Strategic change

This type of Change Request often results in a complete re-development of the architecture, or at least in an iteration of a part of the architecture development cycle, as explained below.

12.5.2 Enterprise Architecture Change Management Process

The Enterprise Architecture change management process needs to determine how changes are to be managed, what techniques are to be applied, and what methodologies used. The process also needs a filtering function that determines which phases of the architecture development process are impacted by requirements. For example, changes that affect only migration may be of no interest in the architecture development phases.

There are many valid approaches to change management, and various management techniques and methodologies that can be used to manage change; for example, project management methods such as PRINCE2, service management methods such as ITIL, management consultancy methods such as Catalyst, and many others.

An enterprise that already has a change management process in place in a field other than

architecture (for example, in systems development or project management) may well be able to adapt it for use in relation to architecture.

The following describes an approach to change management, aimed particularly at the support of a dynamic Enterprise Architecture, which may be considered for use if no similar process currently exists.

The approach is based on classifying required architectural changes into one of three categories:

- **Simplification change:** a simplification change can normally be handled via change management techniques
- **Incremental change:** an incremental change may be capable of being handled via change management techniques, or it may require partial re-architecting, depending on the nature of the change (see [Section 12.5.3](#) for guidelines)
- **Re-architecting change:** a re-architecting change requires putting the whole architecture through the architecture development cycle again

Another way of looking at these three choices is to say that a simplification change to an architecture is often driven by a requirement to reduce investment; an incremental change is driven by a requirement to derive additional value from existing investment; and a re-architecting change is driven by a requirement to increase investment in order to create new value for exploitation.

To determine whether a change is simplification, incremental, or re-architecting, the following activities are undertaken:

1. Registration of all events that may impact the architecture
2. Resource allocation and management for architecture tasks
3. The process or role responsible for architecture resources has to make an assessment of what should be done
4. Evaluation of impacts

12.5.3 Guidelines for Maintenance versus Architecture Redesign

A good guideline is:

- If the change impacts two stakeholders or more, then it is likely to require an architecture redesign and re-entry to the ADM
- If the change impacts only one stakeholder, then it is more likely to be a candidate for change management
- If the change can be allowed under a dispensation, then it is more likely to be a candidate for change management

For example:

- If the impact is significant for the business strategy, then there may be a need to redo the whole Enterprise Architecture — thus a re-architecting approach

- If a new technology or standards emerge, then there may be a need to refresh the Technology Architecture, but not the whole Enterprise Architecture — thus an incremental change
- If the change is at an infrastructure level — for example, ten systems reduced or changed to one system — this may not change the architecture above the physical layer, but it will change the Baseline Description of the Technology Architecture; this would be a simplification change handled via change management techniques

In particular, a refreshment cycle (partial or complete re-architecting) may be required if:

- The Foundation Architecture needs to be re-aligned with the business strategy
- Substantial change is required to components and guidelines for use in deployment of the architecture
- Significant standards used in the product architecture are changed which have significant end-user impact; e.g., regulatory changes

If there is a need for a refreshment cycle, then a new Request for Architecture Work must be issued (to move to another cycle).

Chapter 13: ADM Architecture Requirements Management

This chapter looks at the process of managing architecture requirements throughout the ADM.

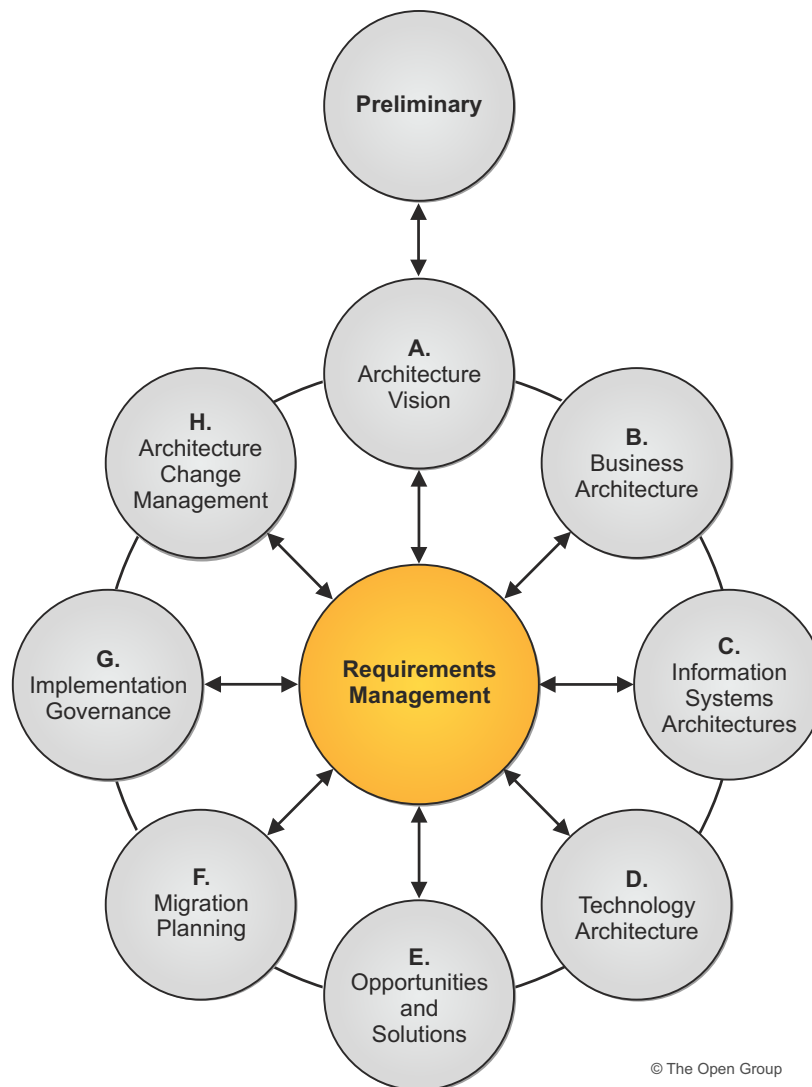


Figure 13-1 ADM Architecture Requirements Management

13.1 Objectives

The objectives of the Requirements Management phase are to:

- Ensure that the Requirements Management process is sustained and operates for all relevant ADM phases
- Manage architecture requirements identified during any execution of the ADM cycle or a phase
- Ensure that relevant architecture requirements are available for use by each phase as the phase is executed

13.2 Inputs

Inputs to the Requirements Management phase are:

- A populated Architecture Repository (see the TOGAF Standard — Architecture Content)
- Organizational Model for Enterprise Architecture (see the TOGAF Standard — Architecture Content), including:
 - Scope of organizations impacted
 - Maturity assessment, gaps, and resolution approach
 - Roles and responsibilities for architecture team(s)
 - Constraints on architecture work
 - Budget requirements
 - Governance and support strategy
- Tailored Architecture Framework (see the TOGAF Standard — Architecture Content), including:
 - Tailored architecture method
 - Tailored architecture content (deliverables and artifacts)
 - Configured and deployed tools
- Statement of Architecture Work (see the TOGAF Standard — Architecture Content)
- Architecture Vision (see the TOGAF Standard — Architecture Content)
- Architecture requirements, populating an Architecture Requirements Specification (see the TOGAF Standard — Architecture Content)
- Requirements Impact Assessment (see the TOGAF Standard — Architecture Content)

13.3 Steps

The steps in the Requirements Management phase are described in the table below:

	Requirements Management Steps	ADM Phase Steps
Step 1		Identify requirements (typically by analyzing how business goals/objectives can be met through the design of value streams, business scenarios, user experiences, or the provision of management information) and document them in the Architecture Requirements Specification and Requirements Repository.
Step 2	Establish baseline requirements: determine priorities, confirm stakeholder agreement to priorities, and document them in the Architecture Requirements Specification and Requirements Repository.	
Step 3	Monitor baseline requirements.	
Step 4		Identify new and changed requirements: <ul style="list-style-type: none"> a. Remove or re-assess priorities b. Add requirements and re-assess priorities c. Modify existing requirements

	Requirements Management Steps	ADM Phase Steps
Step 5	<p>Identify changed requirements and record priorities:</p> <ol style="list-style-type: none"> Identify changed requirements and ensure the requirements are prioritized by the architect(s) responsible for the current phase, and by the relevant stakeholders Record new priorities Ensure that any conflicts are identified and managed through the phases to a successful conclusion and prioritization Generate Requirements Impact Statement (see the TOGAF Standard — Architecture Content) for steering the architecture team <p>Notes</p> <ul style="list-style-type: none"> ■ Changed requirements can come in through any route <p>To ensure that the requirements are properly assessed and prioritized, this process needs to direct the ADM phases and record the decisions related to the requirements.</p> <ul style="list-style-type: none"> ■ The Requirements Management phase needs to determine stakeholder satisfaction with the decisions <p>Where there is dissatisfaction, the phase remains accountable to ensure the resolution of the issues and determine next steps.</p>	

	Requirements Management Steps	ADM Phase Steps
Step 6		<ul style="list-style-type: none"> a. Assess impact of changed requirements on current (active) phase b. Assess impact of changed requirements on previous phases c. Determine whether to implement change, or defer to later ADM cycle; if decision is to implement, assess timescale for change management implementation d. Issue Requirements Impact Statement, Version $n+1$
Step 7		<p>Implement requirements arising from Phase H.</p> <p>The architecture can be changed through its lifecycle by the Architecture Change Management phase (Phase H). The Requirements Management process ensures that new or changing requirements that are derived from Phase H are managed accordingly.</p>
Step 8	Update the Architecture Requirements Repository with information relating to the changes requested, including stakeholder views affected.	
Step 9		Implement change in the current phase.

	Requirements Management Steps	ADM Phase Steps
Step 10		<p>Assess and revise gap analysis for past phases.</p> <p>The gap analysis in the ADM Phases B through D identifies the gaps between Baseline and Target Architectures. Certain types of gap can give rise to gap requirements.</p> <p>The ADM describes two kinds of gap:</p> <ul style="list-style-type: none"> ■ Something that is present in the baseline, but not in the target (i.e., eliminated — by accident or design) ■ Something not in the baseline, but present in the target (i.e., new) <p>A "gap requirement" is anything that has been eliminated by accident, and therefore requires a change to the Target Architecture.</p> <p>If the gap analysis generates gap requirements, then this step will ensure that they are addressed, documented, and recorded in the Architecture Requirements Repository, and that the Target Architecture is revised accordingly.</p>

13.4 Outputs

The outputs of the Requirements Management process may include, but are not restricted to:

- Requirements Impact Assessment (see the TOGAF Standard — Architecture Content)
- Updated Architecture Requirements Specification (see the TOGAF Standard — Architecture Content: Architecture Requirements Specification), if necessary

The TOGAF Standard — Architecture Content contains a detailed description of architectural artifacts which may be produced in this phase, describing them in detail and relating them to entities, attributes, and relationships in the TOGAF Enterprise Metamodel.

The Architecture Requirements Repository will be updated as part of the Requirements Management phase and should contain all requirements information.

When new requirements arise, or existing ones are changed, a Requirements Impact Statement is generated, which identifies the phases of the ADM that need to be revisited to address the changes. The statement goes through various iterations until the final version, which includes the full implications of the requirements (e.g., costs, timescales, and business metrics) on the architecture development. Once requirements for the current ADM cycle have been finalized then the Architecture Requirements Specification should be updated.

13.5 Approach

13.5.1 General

As indicated by the "Requirements Management" circle at the center of the ADM graphic, the ADM is continuously driven by the Requirements Management process.

It is important to note that the Requirements Management circle denotes not a static set of requirements, but a dynamic process whereby requirements for Enterprise Architecture and subsequent changes to those requirements are identified, stored, and fed into and out of the relevant ADM phases, and also between cycles of the ADM.

The ability to deal with changes in requirements is crucial. Architecture is an activity that by its very nature deals with uncertainty and change — the "grey area" between what stakeholders aspire to and what can be specified and engineered as a solution. Architecture requirements are therefore invariably subject to change in practice. Moreover, architecture often deals with drivers and constraints, many of which by their very nature are beyond the control of the enterprise (changing market conditions, new legislation, etc.), and which can produce changes in requirements in an unforeseen manner.

Note also that the Requirements Management process itself does not dispose of, address, or prioritize any requirements; this is done within the relevant phase of the ADM. It is merely the process for managing requirements throughout the overall ADM.

It is recommended that an Architecture Requirements Repository (see the TOGAF Standard — Architecture Content) is used to record and manage all architecture requirements. Unlike the Architecture Requirements Specification, and the Requirements Impact Assessment, the Architecture Requirements Repository can hold information from multiple ADM cycles.

13.5.2 Requirements Development

The first high-level requirements are articulated as part of the Architecture Vision, generated by means of the business scenario or analogous technique.

Each phase of the ADM, from Preliminary to Phase H, must select the approved requirements for that phase as held in the Architecture Requirements Repository and Architecture Requirements Specification. At the completion of the phase the status of all such requirements needs to be updated. During the phase execution, new requirements generated for future architecture work within the scope of the current Statement of Architecture Work need to be documented within the Architecture Requirements Specification, and new requirements which are outside of the scope of the current Statement of Architecture Work must be input to the Architecture Requirements Repository for management through the Requirements Management process.

In each relevant phase of the ADM the architect should identify types of requirement that must be met by the architecture, including applicable:

- Functional requirements
- Non-functional requirements

When defining requirements the architect should take into account:

- Assumptions for requirements
- Constraints for requirements
- Domain-specific principles that drive requirements
- Policies affecting requirements
- Standards that requirements must meet
- Organization guidelines for requirements
- Specifications for requirements

Deliverables in later ADM phases also contain mappings to the design requirements, and may also generate new types of requirements (for example, conformance requirements, time windows for implementation).

13.5.3 Resources

The world of requirements engineering is rich with emerging recommendations and processes for Requirements Management. The TOGAF Standard does not mandate or recommend any specific process or tool; it simply states what an effective Requirements Management process should achieve (i.e., the "requirements for requirements", if you like).

13.5.3.1 Business Scenarios

A technique used to analyze how a business goal or objective can be met by a process or value stream. Analyzing where the activities in that process are performed by human and computer actors is a highly effective way to identify and clarify architecture requirements. The technique is detailed in the TOGAF® Series Guide: Business Scenarios.

13.5.3.2 Requirements Tools

There is a large, and increasing, number of Commercial Off-The-Shelf (COTS) tools available for the support of Requirements Management, albeit not necessarily designed for architecture requirements.

Index

activity model	53
ADM	1, 24, 125
Requirements Management	129
ADM guidelines and techniques	2
Application Architecture	71
application building blocks	74
Architecture Board	7
architecture change management	119
Architecture Development Method	1
architecture domain	8
architecture domains	11
Architecture Governance	7, 24, 117
architecture integration	13
architecture models	74, 84
architecture partition	8
Architecture Principles	24
Architecture Report	78
Architecture Repository	55, 69, 91
Architecture Vision	5, 29
architecture, scoping	7
audit information	7
Balanced Scorecard technique	20
Baseline Application Architecture Description	76
Baseline Data Architecture Description	64
Baseline Description	51
Baseline Technology Architecture Description	87
BPMN	54
breadth, enterprise	8-9
Business Architecture	41
business capabilities	39, 51
business capability mapping	45
business planning	26
business principles	24
business process model	53
Business Process Modeling Notation	54
business scenarios	136
catalog	85
catalog of application building blocks	74
catalogs, building blocks	45
Catalyst	126
change management	119
change management process	126
class model	54
COBIT	7

content tailoring	20
COTS	136
create architecture models	62
Data Architecture	59
data building blocks	62
data entity	67
data governance	69
data management	68
data migration	68
data structure	67
depth, enterprise	8-9
diagrams	63, 75, 86
drivers for change	125
EAP	58
emerging technologies	91
enterprise	22
formal stakeholder review	48, 65, 78, 89
gap analysis	77, 88
ICAM	54
ICOM	53
IDEF	54
implementation governance	111
incremental change	127
Industry Architectures	55
information mapping	45, 53
Information Systems Architecture	57
ITIL	126
lessons learned process	126
logical data component	67
logical data model	54
management frameworks	26
matrices	63, 75, 86
MDA	73
Migration Planning	103
Model-Driven Architecture	73
modeling process	44
modeling techniques	53
NASCIO	27
OMG	54, 73
operations management	27
Opportunities & Solutions	93
organization map	39, 52
organization mapping	45
Phase A	29
Phase B	41
Phase C	57, 59, 71
Phase D	81
Phase E	93
Phase F	103
Phase G	111

Phase H	119
physical data component	68
portfolio management	27
Preliminary Phase.....	15
PRINCE2	126
process modeling.....	45
process status.....	7
process tailoring.....	19
Product Line Architecture	6
project management	27
re-architecting change	127
reference data.....	7
reference models	61, 73, 84
Request for Architecture Work.....	37
Requirement for Architecture Work.....	24
requirements	63, 76
requirements tools	136
roadmap components	47, 77, 89
scoping the architecture.....	7
simplification change	127
SOA	13
Statement of Architecture Work.....	40
structured analysis.....	45
Target Architecture Description.....	10
Technology Architecture.....	81
technology building blocks	85
terminology tailoring.....	19
time period	10
time period, enterprise.....	8
TOGAF ADM.....	125
tools	61, 73, 84
tools standardization.....	20
Transition Architecture Description	10
UML	54
use-case analysis	45
use-case model	54
value stream mapping.....	45
value streams.....	39, 52
viewpoints	61, 73, 84
Zachman Framework	5

The Open Group Standard

TOGAF® Standard — ADM Techniques

The Open Group

Copyright © 2005-2022, The Open Group

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

Any use of this publication for commercial purposes is subject to the terms of the Annual Commercial License relating to it. For further information, see www.opengroup.org/legal/licensing.

The Open Group Standard

TOGAF® Standard — ADM Techniques

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

Chapter	1	Introduction	1
Chapter	2	Architecture Principles	3
	2.1	Introduction	3
	2.2	Characteristics of Architecture Principles	4
	2.3	Components of Architecture Principles	4
	2.4	Developing Architecture Principles	5
	2.4.1	Qualities of Principles	5
	2.5	Applying Architecture Principles	6
	2.6	Example Set of Architecture Principles	7
	2.6.1	Business Principles	7
	2.6.2	Data Principles	11
	2.6.3	Application Principles	16
	2.6.4	Technology Principles	17
Chapter	3	Stakeholder Management	21
	3.1	Introduction	21
	3.2	Approach to Stakeholder Management	21
	3.3	Steps in the Stakeholder Management Process	22
	3.3.1	Identify Stakeholders	22
	3.3.2	Classify Stakeholder Positions	24
	3.3.3	Determine Stakeholder Management Approach	24
	3.3.4	Tailor Engagement Deliverables	25
	3.4	Template Stakeholder Map	25
Chapter	4	Architecture Patterns	39
	4.1	Introduction	39
	4.1.1	Background	39
	4.1.2	Content of a Pattern	40
	4.1.3	Terminology	41
	4.2	Some Pattern Resources	42
Chapter	5	Gap Analysis	45
	5.1	Introduction	45
	5.2	Suggested Steps	46
	5.3	Example	46
Chapter	6	Migration Planning Techniques	49
	6.1	Implementation Factor Catalog	49
	6.2	Consolidated Gaps, Solutions, & Dependencies Matrix	50
	6.3	Architecture Definition Increments Table	50
	6.4	Transition Architecture State Evolution Table	51

	6.5	Business Value Assessment Technique	52
Chapter	7	Interoperability Requirements	53
	7.1	Overview.....	53
	7.2	Defining Interoperability	53
	7.3	Enterprise Operating Model	55
	7.4	Refining Interoperability	55
	7.5	Determining Interoperability Requirements	56
	7.6	Reconciling Interoperability Requirements with Potential Solutions	58
Chapter	8	Business Transformation Readiness Assessment.....	59
	8.1	Introduction.....	59
	8.1.1	Business Transformation Enablement Program (BTEP).....	60
	8.2	Determine Readiness Factors	60
	8.3	Present Readiness Factors	62
	8.4	Assess Readiness Factors	63
	8.4.1	Readiness Factor Vision.....	64
	8.4.2	Readiness Factor Rating	64
	8.4.3	Readiness Factor Risks & Actions.....	65
	8.5	Readiness and Migration Planning	66
	8.6	Marketing the Implementation Plan	66
	8.7	Conclusion.....	66
Chapter	9	Risk Management.....	67
	9.1	Introduction.....	67
	9.2	Risk Classification	68
	9.3	Risk Identification	68
	9.4	Initial Risk Assessment.....	68
	9.5	Risk Mitigation and Residual Risk Assessment.....	70
	9.6	Conduct Residual Risk Assessment.....	70
	9.7	Risk Monitoring and Governance (Phase G).....	70
	9.8	Summary	71
Chapter	10	Architecture Alternatives and Trade-Offs.....	73
	10.1	Concept	73
	10.2	Method.....	73
	10.2.1	Criteria	74
	10.2.2	Identify Alternatives	75
	10.2.3	Choose from Alternatives and Define in Detail.....	76
		Index	77
List of Figures			
	3-1	Sample Stakeholders and Categories	23
	3-2	Stakeholder Power Grid.....	24
	5-1	Gap Analysis Example	46
	6-1	Implementation Factor Catalog.....	49
	6-2	Consolidated Gaps, Solutions, and Dependencies Matrix	50

Contents

6-3	Architecture Definition Increments Table	50
6-4	Transition Architecture State Evolution Table	51
6-5	Sample Project Assessment with Respect to Business Value and Risk.....	52
7-1	Business Information Interoperability Matrix.....	57
7-2	Information Systems Interoperability Matrix	57
8-1	Business Transformation Readiness Assessment — Maturity Model	63
8-2	Summary Table of Business Transformation Readiness Assessment	64
9-1	Risk Classification Scheme	69
9-2	Sample Risk Identification and Mitigation Assessment Worksheet	70
10-1	Architecture Trade-Off Method	73

List of Tables

2-1	Recommended Format for Defining Principles	4
3-1	Example Stakeholder Analysis	24

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.

Preface

This Document

This is the TOGAF Standard — ADM Techniques.

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.

FICO is a registered trademark of Fair Isaac Corporation in the United States and other countries.

Java is a registered trademark of Oracle and/or its affiliates.

PMBOK is a registered trademark of the Project Management Institute, Inc. which is registered in the United States and other nations.

PRINCE2 is a registered trademark of AXELOS Limited.

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.

Referenced Documents

Chapter 1: Introduction

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

Guidelines included within this document are as follows:

- Architecture Principles (see [Chapter 2](#)) describes principles for the use and deployment of IT resources across the enterprise, and how to develop the set of general rules and guidelines for the architecture being developed
- Stakeholder Management (see [Chapter 3](#)) describes stakeholder management, an important discipline that successful architecture practitioners can use to win support for their projects
- Architecture Patterns (see [Chapter 4](#)) provides guidance on using architectural patterns
- Gap Analysis (see [Chapter 5](#)) 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 [Chapter 6](#)) describes a number of techniques to support migration planning in Phases E and F
- Interoperability Requirements (see [Chapter 7](#)) describes a technique for determining interoperability requirements
- Business Transformation Readiness Assessment (see [Chapter 8](#)) describes a technique for identifying business transformation issues
- Risk Management (see [Chapter 9](#)) describes a technique for managing risk during an architecture/business transformation project
- Architecture Alternatives and Trade-Offs (see [Chapter 10](#)) describes a technique to identify alternative Target Architectures and perform trade-offs between the alternatives

Chapter 2: Architecture Principles

This chapter describes principles for use in the development of an Enterprise Architecture.

2.1 Introduction

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

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

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

- **Enterprise Principles** provide a basis for decision-making throughout an enterprise, and inform how the organization sets about fulfilling its mission

Such principles are commonly found as a means of harmonizing decision-making across an organization. In particular, they are a key element in a successful Architecture Governance strategy (see the TOGAF Standard — Enterprise Architecture Capability and Governance).

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

- **Architecture Principles** are a set of principles that relate to architecture work

They reflect a level of consensus across the enterprise, and embody the spirit and thinking of existing enterprise principles. Architecture Principles govern the architecture process, affecting the development, maintenance, and use of the Enterprise Architecture.

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

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

The remainder of this section deals exclusively with Architecture Principles.

2.2 Characteristics of Architecture Principles

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

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

2.3 Components of Architecture Principles

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

A recommended template is given in [Table 2-1](#).

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

Table 2-1 Recommended Format for Defining Principles

An example set of Architecture Principles following this template is given in [Section 2.6](#).

2.4 Developing Architecture Principles

Architecture Principles are typically developed by the Enterprise Architects, in conjunction with the key stakeholders, and are approved by the Architecture Board.

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

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

Specifically, the development of Architecture Principles is typically influenced by the following:

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

2.4.1 Qualities of Principles

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

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

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

- **Understandable:** the underlying tenets can be quickly grasped and understood by individuals throughout the organization

The intention of the principle is clear and unambiguous, so that violations, whether intentional or not, are minimized.
- **Robust:** enable good quality decisions about architectures and plans to be made, and enforceable policies and standards to be created

Each principle should be sufficiently definitive and precise to support consistent decision-making in complex, potentially controversial situations.

- **Complete:** every potentially important principle governing the management of information and technology for the organization is defined — the principles cover every situation perceived

- **Consistent:** strict adherence to one principle may require a loose interpretation of another principle

The set of principles must be expressed in a way that allows a balance of interpretations. Principles should not be contradictory to the point where adhering to one principle would violate the spirit of another. Every word in a principle statement should be carefully chosen to allow consistent yet flexible interpretation.

- **Stable:** principles should be enduring, yet able to accommodate changes

An amendment process should be established for adding, removing, or altering principles after they are ratified initially.

2.5 Applying Architecture Principles

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

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

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

Principles will sometimes compete; for example, the principles of "accessibility" and "security"

tend towards conflicting decisions. Each principle must be considered in the context of "all other things being equal".

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

A common reaction on first reading of a principle is "this is obvious and does not need to be documented". The fact that a principle seems self-evident does not mean that the guidance in a principle is followed. Having principles that appear obvious helps ensure that decisions actually follow the desired outcome.

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

2.6 Example Set of Architecture Principles

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

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

2.6.1 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.

Implications:

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

Principle 2: Maximize Benefit to the Enterprise

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

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

- Implications:
- Achieving maximum enterprise-wide benefit will require changes in the way we plan and manage information — technology alone will not bring about this change
 - Some organizations may have to concede their own preferences for the greater benefit of the entire enterprise
 - Where feasible, application development priorities must be established by the entire enterprise for the entire enterprise
 - Applications components should be shared across organizational boundaries
 - Information management initiatives should be conducted in accordance with the enterprise plan
- Individual organizations should pursue information management initiatives which conform to the blueprints and priorities established by the enterprise. The plan will be changed as needed.
- As needs arise, priorities must be adjusted; a forum with comprehensive enterprise representation should make these decisions

Principle 3: Information Management is Everybody's Business

- Statement: All organizations in the enterprise participate in information management decisions needed to accomplish business objectives.
- Rationale: Information users are the key stakeholders, or customers, in the application of technology to address a business need. In order to ensure information management is aligned with the business, all organizations in the enterprise must be involved in all aspects of the information environment. The business experts from across the enterprise and the technical staff responsible for developing and sustaining the information environment need to come together as a team to jointly define the goals and objectives of IT.
- Implications:
- To operate as a team, every stakeholder, or customer, will need to accept responsibility for developing the information environment
 - Commitment of resources will be required to implement this principle

Principle 4: Business Continuity

- Statement: Enterprise operations are maintained in spite of system interruptions.
- Rationale: As system operations become more pervasive, we become more dependent on them; therefore, we must consider the reliability of such systems throughout their design and use. Business premises throughout the enterprise must be provided with the capability to continue operations regardless of external events. Hardware failure, natural disasters, and data corruption should not be allowed to disrupt or stop enterprise activities. The enterprise must be capable of operating on alternative information delivery mechanisms.
- Implications:
- Dependency on shared system applications mandates that the risks of business interruption must be established in advance and managed
- Management includes but is not limited to periodic reviews, testing for

vulnerability and exposure, or designing mission-critical services to ensure business continuity through redundant or alternative capabilities.

- Recoverability, redundancy, and maintainability should be addressed at the time of design
- Applications must be assessed for criticality and impact on the enterprise mission, in order to determine what level of continuity is required and what corresponding recovery plan is necessary

Principle 5: Common Use Applications

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

Rationale: Duplicative capability is expensive and proliferates conflicting data.

Implications:

- Organizations which depend on a capability which does not serve the entire enterprise must change over to the replacement enterprise-wide capability; this will require establishment of and adherence to a policy requiring this
- Organizations will not be allowed to develop capabilities for their own use which are similar/duplicative of enterprise-wide capabilities; in this way, expenditures of scarce resources to develop essentially the same capability in marginally different ways will be reduced
- Data and information used to support enterprise decision-making will be standardized to a much greater extent than previously

This is because the smaller, organizational capabilities which produced different data (which was not shared among other organizations) will be replaced by enterprise-wide capabilities. The impetus for adding to the set of enterprise-wide capabilities may well come from an organization making a convincing case for the value of the data/information previously produced by its organizational capability, but the resulting capability will become part of the enterprise-wide system, and the data it produces will be shared across the enterprise.

Principle 6: Service Orientation

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

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

Implications:

- Service representation utilizes business descriptions to provide context (i.e., business process, goal, rule, policy, service interface, and service component) and implements services using service orchestration
- Service orientation places unique requirements on the infrastructure, and implementations should use open standards to realize interoperability and location transparency

- Implementations are environment-specific; they are constrained or enabled by context and must be described within that context
- Strong governance of service representation and implementation is required
- A "Litmus Test", which determines a "good service", is required

Principle 7: Compliance with Law

Statement:	Enterprise information management processes comply with all relevant laws, policies, and regulations.
Rationale:	Enterprise policy is to abide by laws, policies, and regulations. This will not preclude business process improvements that lead to changes in policies and regulations.
Implications:	<ul style="list-style-type: none">■ The enterprise must be mindful to comply with laws, regulations, and external policies regarding the collection, retention, and management of data■ Education and access to the rules <p>Efficiency, need, and common sense are not the only drivers. Changes in the law and changes in regulations may drive changes in our processes or applications.</p>

Principle 8: IT Responsibility

Statement:	The IT organization is responsible for owning and implementing IT processes and infrastructure that enable solutions to meet user-defined requirements for functionality, service levels, cost, and delivery timing.
Rationale:	Effectively align expectations with capabilities and costs so that all projects are cost-effective. Efficient and effective solutions have reasonable costs and clear benefits.
Implications:	<ul style="list-style-type: none">■ A process must be created to prioritize projects■ The IT function must define processes to manage business unit expectations■ Data, application, and technology models must be created to enable integrated quality solutions and to maximize results

Principle 9: Protection of Intellectual Property

Statement:	The enterprise's Intellectual Property (IP) must be protected. This protection must be reflected in the IT architecture, implementation, and governance processes.
Rationale:	A major part of an enterprise's IP is hosted in the IT domain.
Implications:	<ul style="list-style-type: none">■ While protection of IP assets is everybody's business, much of the actual protection is implemented in the IT domain — even trust in non-IT processes can be managed by IT processes (email, mandatory notes, etc.)

- A security policy, governing human and IT actors, will be required that can substantially improve protection of IP; this must be capable of both avoiding compromises and reducing liabilities
- Resources on such policies can be found at the SANS Institute (refer to www.sans.org/security-resources/policies)

2.6.2 Data Principles

Principle 10: Data is an Asset

Statement:	Data is an asset that has value to the enterprise and is managed accordingly.
Rationale:	Data is a valuable corporate resource; it has real, measurable value. In simple terms, the purpose of data is to aid decision-making. Accurate, timely data is critical to accurate, timely decisions. Most corporate assets are carefully managed, and data is no exception. Data is the foundation of our decision-making, so we must also carefully manage data to ensure that we know where it is, can rely upon its accuracy, and can obtain it when and where we need it.
Implications:	<ul style="list-style-type: none"> ■ This is one of three closely-related principles regarding data: data is an asset; data is shared; and data is easily accessible <p>The implication is that there is an education task to ensure that all organizations within the enterprise understand the relationship between value of data, sharing of data, and accessibility to data.</p> <ul style="list-style-type: none"> ■ Stewards must have the authority and means to manage the data for which they are accountable ■ We must make the cultural transition from "data ownership" thinking to "data stewardship" thinking ■ The role of data steward is critical because obsolete, incorrect, or inconsistent data could be passed to enterprise personnel and adversely affect decisions across the enterprise ■ Part of the role of data steward, who manages the data, is to ensure data quality <p>Procedures must be developed and used to prevent and correct errors in the information and to improve those processes that produce flawed information. Data quality will need to be measured and steps taken to improve data quality — it is probable that policy and procedures will need to be developed for this as well.</p> <ul style="list-style-type: none"> ■ A forum with comprehensive enterprise-wide representation should decide on process changes suggested by the steward ■ Since data is an asset of value to the entire enterprise, data stewards accountable for properly managing the data must be assigned at the enterprise level

Principle 11: Data is Shared

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

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

Shared data will result in improved decisions since we will rely on fewer (ultimately one virtual) sources of more accurate and timely managed data for all of our decision-making. Electronically shared data will result in increased efficiency when existing data entities can be used, without re-keying, to create new entities.

Implications:

- This is one of three closely-related principles regarding data: data is an asset; data is shared; and data is easily accessible

The implication is that there is an education task to ensure that all organizations within the enterprise understand the relationship between value of data, sharing of data, and accessibility to data.

- To enable data sharing we must develop and abide by a common set of policies, procedures, and standards governing data management and access for both the short and the long term
- For the short term, to preserve our significant investment in legacy systems, we must invest in software capable of migrating legacy system data into a shared data environment
- We will also need to develop standard data models, data elements, and other metadata that defines this shared environment and develop a repository system for storing this metadata to make it accessible
- For the long term, as legacy systems are replaced, we must adopt and enforce common data access policies and guidelines for new application developers to ensure that data in new applications remains available to the shared environment and that data in the shared environment can continue to be used by the new applications
- For both the short term and the long term we must adopt common methods and tools for creating, maintaining, and accessing the data shared across the enterprise
- Data sharing will require a significant cultural change
- This principle of data sharing will continually "bump up against" the principle of data security — under no circumstances will the data sharing principle cause confidential data to be compromised

- Data made available for sharing will have to be relied upon by all users to execute their respective tasks

This will ensure that only the most accurate and timely data is relied upon for decision-making. Shared data will become the enterprise-wide "virtual single source" of data.

Principle 12: Data is Accessible

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

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

Implications: ■ This is one of three closely-related principles regarding data: data is an asset; data is shared; and data is easily accessible

The implication is that there is an education task to ensure that all organizations within the enterprise understand the relationship between value of data, sharing of data, and accessibility to data.

- Accessibility involves the ease with which users obtain information
- The way information is accessed and displayed must be sufficiently adaptable to meet a wide range of enterprise users and their corresponding methods of access
- Access to data does not constitute understanding of the data — personnel should take caution not to misinterpret information
- Access to data does not necessarily grant the user access rights to modify or disclose the data

This will require an education process and a change in the organizational culture, which currently supports a belief in "ownership" of data by functional units.

Principle 13: Data Trustee

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

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

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

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

Principle 14: Common Vocabulary and Data Definitions

- Statement: Data is defined consistently throughout the enterprise, and the definitions are understandable and available to all users.
- Rationale: The data that will be used in the development of applications must have a common definition throughout the Headquarters to enable sharing of data. A common vocabulary will facilitate communications and enable dialog to be effective. In addition, it is required to interface systems and exchange data.
- Implications:
- We are lulled into thinking that this issue is adequately addressed because there are people with "data administration" job titles and forums with charters implying responsibility
Significant additional energy and resources must be committed to this task. It is key to the success of efforts to improve the information environment. This is separate from but related to the issue of data element definition, which is addressed by a broad community — this is more like a common vocabulary and definition.
 - The enterprise must establish the initial common vocabulary for the business; the definitions will be used uniformly throughout the enterprise
 - Whenever a new data definition is required, the definition effort will be co-ordinated and reconciled with the corporate "glossary" of data descriptions
The enterprise data administrator will provide this co-ordination.

- Ambiguities resulting from multiple parochial definitions of data must give way to accepted enterprise-wide definitions and understanding
- Multiple data standardization initiatives need to be co-ordinated
- Functional data administration responsibilities must be assigned

Principle 15: Data Security

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

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

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

Implications: ■ Aggregation of data, both classified and not, will create a large target requiring review and de-classification procedures to maintain appropriate control

Data owners and/or functional users must determine whether the aggregation results in an increased classification level. Appropriate policy and procedures will be needed to handle this review and de-classification. Access to information based on a need-to-know policy will force regular reviews of the body of information.

- The current practice of having separate systems to contain different classifications needs to be rethought

Is there a software solution to separating classified and unclassified data? The current hardware solution is unwieldy, inefficient, and costly. It is more expensive to manage unclassified data on a classified system. Currently, the only way to combine the two is to place the unclassified data on the classified system, where it must remain.

- In order to adequately provide access to open information while maintaining secure information, security needs must be identified and developed at the data level, not the application level

- Data security safeguards can be put in place to restrict access to "view only" or "never see"

Sensitivity labeling for access to pre-decisional, decisional, classified, sensitive, or proprietary information must be determined.

- Security must be designed into data elements from the beginning; it cannot be added later

Systems, data, and technologies must be protected from unauthorized access and manipulation. Information at Headquarters must be

safeguarded against inadvertent or unauthorized alteration, sabotage, disaster, or disclosure.

- New policies are needed on managing duration of protection for pre-decisional information and other works-in-progress, in consideration of content freshness

2.6.3 Application Principles

Principle 16: Technology Independence

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

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

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

- Implications:**
- This principle will require standards which support portability
 - For Commercial Off-The-Shelf (COTS) and Government Off-The-Shelf (GOTS) applications, there may be limited current choices, as many of these applications are technology and platform-dependent
 - Subsystem interfaces will need to be developed to enable legacy applications to interoperate with applications and operating environments developed under the Enterprise Architecture
 - Middleware should be used to decouple applications from specific software solutions
 - As an example, this principle could lead to use of Java[®], and future Java-like protocols, which give a high degree of priority to platform-independence

Principle 17: Ease-of-Use

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

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

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

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

2.6.4 Technology Principles

Principle 18: Requirements-Based Change

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

Rationale: This principle will foster an atmosphere where the information environment changes in response to the needs of the business, rather than having the business change in response to IT changes. This is to ensure that the purpose of the information support — the transaction of business — is the basis for any proposed change.

Unintended effects on business due to IT changes will be minimized.

A change in technology may provide an opportunity to improve the business process and, hence, change business needs.

- Implications:
- Changes in implementation will follow full examination of the proposed changes using the Enterprise Architecture
 - There is no funding for a technical improvement or system development unless a documented business need exists
 - Change management processes conforming to this principle will be developed and implemented
 - This principle may bump up against the responsive change principle
- We must ensure the requirements documentation process does not hinder responsive change to meet legitimate business needs. The purpose of this principle is to keep the focus on business, not technology needs — responsive change is also a business need.

Principle 19: Responsive Change Management

- Statement: Changes to the enterprise information environment are implemented in a timely manner.
- Rationale: If people are to be expected to work within the enterprise information environment, that information environment must be responsive to their needs.
- Implications:
- Processes for managing and implementing change must be developed that do not create delays
 - A user who feels a need for change will need to connect with a "business expert" to facilitate explanation and implementation of that need
 - If changes are going to be made, the architecture must be kept updated
 - Adopting this principle might require additional resources
 - This will conflict with other principles (e.g., maximum enterprise-wide benefit, enterprise-wide applications, etc.)

Principle 20: Control Technical Diversity

- Statement: Technological diversity is controlled to minimize the non-trivial cost of maintaining expertise in and connectivity between multiple processing environments.
- Rationale: There is a real, non-trivial cost of infrastructure required to support alternative technologies for processing environments. There are further infrastructure costs incurred to keep multiple processor constructs interconnected and maintained.
- Limiting the number of supported components will simplify maintainability and reduce costs.
- The business advantages of minimum technical diversity include: standard packaging of components; predictable implementation impact; predictable valuations and returns; redefined testing; utility status; and increased flexibility to accommodate technological advancements. Common technology across the enterprise brings the benefits of economies of scale to the enterprise. Technical administration and support costs are better controlled when limited resources can focus on this shared set of technology.
- Implications:
- Policies, standards, and procedures that govern the acquisition of technology must be tied directly to this principle
 - Technology choices will be constrained by the choices available within the technology blueprint
- Procedures for augmenting the acceptable technology set to meet evolving requirements will have to be developed and put in place.
- The technology baseline is not being frozen
- Technology advances are welcomed and will change the technology blueprint when compatibility with the current infrastructure, improvement in operational efficiency, or a required capability has been demonstrated.

Principle 21: Interoperability

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

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

Implications:

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

Chapter 3: Stakeholder Management

3.1 Introduction

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

The benefits of successful stakeholder management are that:

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

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

3.2 Approach to Stakeholder Management

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

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

For example, just as a building architect will create wiring diagrams, floor plans, and elevations to describe different facets of a building to its different stakeholders (electricians, owners, planning

officials), so an Enterprise Architect must create different architecture views of the Business, Information Systems, and Technology Architecture for the stakeholders who have concerns related to these aspects.

The TOGAF Standard specifically identifies this issue throughout the ADM through the following concepts (see the TOGAF Standard — Architecture Content):

- Architecture View
- Architecture Viewpoint
- Concern
- Stakeholder

3.3 Steps in the Stakeholder Management Process

The following sections detail recommended stakeholder management activity.

3.3.1 Identify Stakeholders

Identify the key stakeholders of the Enterprise Architecture.

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

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

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

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

Look at who is impacted by the Enterprise Architecture project:

- Who gains and who loses from this change?
- Who controls change management of processes?
- Who designs new systems?
- Who will make the decisions?
- Who procures IT systems and who decides what to buy?
- Who controls resources?
- Who has specialist skills the project needs?
- Who has influence?

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

position of power.

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

3.3.1.1 Sample Stakeholder Analysis

A sample stakeholder analysis that distinguishes 22 types of stakeholder, in five broad categories, is shown in [Figure 3-1](#). Any particular architecture project may have more, fewer, or different stakeholders; and they may be grouped into more, fewer, or different categories.

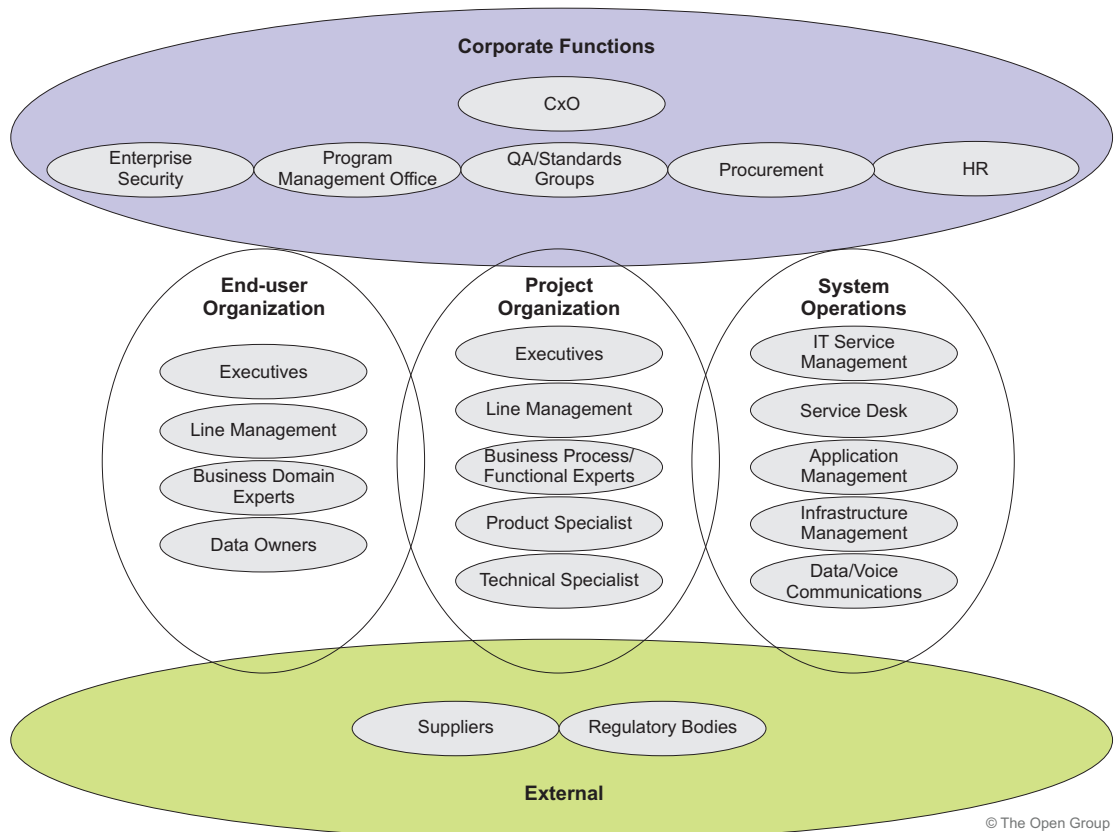


Figure 3-1 Sample Stakeholders and Categories

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

3.3.2 Classify Stakeholder Positions

Develop a good understanding of the most important stakeholders and record this analysis for reference and refresh during the project. An example stakeholder analysis is shown in [Table 3-1](#).

Stakeholder Group	Stakeholder	Ability to Disrupt Change	Current Understanding	Required Understanding	Current Commitment	Required Commitment	Required Support
CIO	John Smith	H	M	H	L	M	H
CFO	Jeff Brown	M	M	M	L	M	M

Table 3-1 Example Stakeholder Analysis

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

This can be done by asking a series of questions:

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

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

3.3.3 Determine Stakeholder Management Approach

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

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

Work out stakeholder power, influence, and interest, so as to focus the Enterprise Architecture engagement on the key individuals. These can be mapped onto a power/interest matrix, which also indicates the strategy to adopt for engaging with them. [Figure 3-2](#) shows an example power grid matrix.

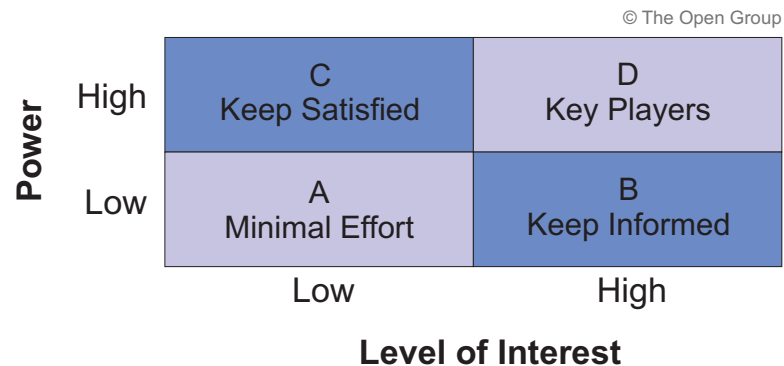


Figure 3-2 Stakeholder Power Grid

3.3.4 Tailor Engagement Deliverables

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

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

3.4 Template Stakeholder Map

The following table provides an example stakeholder map for a TOGAF architecture project which has stakeholders as identified in [Figure 3-1](#).

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

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Program Management Office (Corporate Functions); e.g., Project Portfolio Managers	Prioritizing, funding, and aligning change activity. An understanding of project content and technical dependencies between projects supports portfolio management decision-making.	KEEP SATISFIED	Requirements catalog Project Context diagram Benefits diagram Business Footprint diagram Application Communication diagram Organization map Business Capabilities catalog Capability/Organization matrix Business Capability Map Strategy/Capability matrix Capability/Organization matrix Business Model diagram Value Stream catalog Value Stream Stages catalog Value Stream/Capability matrix Value Stream Map
Procurement (Corporate Functions); e.g., Acquirers	Understanding what building blocks of the architecture can be bought, and what constraints (or rules) are relevant to the purchase. Acquirers will shop with multiple vendors looking for the best cost solution while adhering to the constraints (or rules) derived from the architecture, such as standards. The key concern is to make purchasing decisions that fit the architecture.	KEY PLAYERS	Technology Portfolio catalog Technology Standards catalog

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Human Resources (HR) (Corporate Functions); e.g., HR Managers, Training & Development Managers	The roles and actors are required to support the architecture and changes to it. The key concern is managing people transitions.	KEEP INFORMED	<p>Organization Decomposition diagram</p> <p>Organization/Actor catalog</p> <p>Location catalog</p> <p>Application and User Location diagram</p> <p>Business Capabilities catalog</p> <p>Capability/Organization matrix</p> <p>Business Capability Map</p> <p>Strategy/Capability matrix</p> <p>Capability/Organization matrix</p> <p>Business Model diagram</p>
Enterprise Security (Corporate Functions); e.g., Corporate Risk Management, Security Officers, IT Security Managers	Ensuring that the information, data, and systems of the organization are available to only those that have permission, and protecting the information, data, and systems from unauthorized tampering.	KEY PLAYERS	<p>Product Lifecycle diagram</p> <p>Data Dissemination diagram</p> <p>Data Security diagram</p> <p>Actor/Role matrix</p> <p>Networked Computing Hardware diagram</p> <p>Network and Communications diagram</p>

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
QA/Standards Group (Corporate Functions); e.g., Data Owners, Process Owners, Technical Standards Bodies	Ensuring the consistent governance of the organization's business, data, application, and technology assets.	KEY PLAYERS	Process/Event/Control/Product catalog Contract/Measure catalog Application Portfolio catalog Interface catalog Technology Standards catalog Technology Portfolio catalog Value Stream catalog Value Stream Stages catalog Value Stream/Capability matrix Value Stream Map

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Executive (End-user Organization); e.g., Business Unit Directors, Business Unit CxOs, Business Unit Head of IT/Architecture	The high-level drivers, goals, and objectives of the organization, and how these are translated into an effective process and architecture to advance the business.	KEEP SATISFIED	Business Footprint diagram Goal/Objective/Business Service diagram Organization Decomposition diagram Process Flow diagram Application Communication diagram Business Capabilities catalog Capability/Organization matrix Business Capability Map Strategy/Capability matrix Capability/Organization matrix Business Model diagram Value Stream catalog Value Stream Stages catalog Value Stream/Capability matrix Value Stream Map

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Line Management (End-user Organization); e.g., Senior Business Managers, Operations Regional Managers, IT Managers	Top-level functions and processes of the organization, and how the key applications support these processes.	KEY PLAYERS	Business Footprint diagram Organization Decomposition diagram Organization map Process Flow diagram Application Communication diagram Application and User Location diagram Business Capabilities catalog Capability/Organization matrix Business Capability Map Strategy/Capability matrix Capability/Organization matrix Business Model diagram Value Stream catalog Value Stream Stages catalog Value Stream/Capability matrix Value Stream Map

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Business Domain Experts (End-user Organization); e.g., Business Process Experts, Business/Process Analyst, Process Architect, Process Designer, Functional Managers, Business Analyst	Functional aspects of processes and supporting systems. This can cover the human actors involved in the system, the user processes involved in the system, the functions required to support the processes, and the information required to flow in support of the processes.	KEY PLAYERS	Business Interaction matrix Actor/Role matrix Business Service/Information diagram Organization map Product Lifecycle diagram Business Use-Case diagram Application Use-Case diagram Application Communication diagram Data Entity/Business Function matrix Value Stream catalog Value Stream Stages catalog Value Stream/Capability matrix Value Stream Map
IT Service Management (Systems Operations); e.g., Service Delivery Manager	Ensuring that IT services provided to the organization meet the service levels required by that organization to succeed in business.	KEEP INFORMED	Technology Standards catalog Technology Portfolio catalog Contract/Measure catalog Process/Application Realization diagram Enterprise Manageability diagram

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
IT Operations — Applications (System Operations); e.g., Application Architecture, System & Software Engineers	Development approach, software modularity and re-use, portability migration, and interoperability.	KEY PLAYERS	Process/Application Realization diagram Application/Data matrix Application Migration diagram Software Engineering diagram Platform decomposition Diagram Networked Computing/ Hardware diagram Software distribution Diagram
IT Operations — Infrastructure (System Operations); e.g., Infrastructure Architect, Wintel support, Mid-range support, Operational DBA, Service Desk	Location, modifiability, re-usability, and availability of all components of the system. Ensuring that the appropriate components are developed and deployed within the system in an optimal manner.	KEY PLAYERS	Platform Decomposition diagram Technology Standards catalog Technology Portfolio catalog Enterprise Manageability diagram Networked Computing/ Hardware diagram Processing diagram Environments and Locations diagram
IT Operations — Data/Voice Communications (System Operations); e.g., Network Management	Location, modifiability, re-usability, and availability of communications and networking services. Ensuring that the appropriate communications and networking services are developed and deployed within the system in an optimal manner.	KEY PLAYERS	Network and Communications diagram

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Executive (Project Organization); e.g., Sponsor, Program Manager	On-time, on-budget delivery of a change initiative that will realize expected benefits for the organization.	KEEP INFORMED	Requirements catalog Principles catalog Value Chain diagram Solution Concept diagram Organization map Application and User Location diagram Business Capabilities catalog Capability/Organization matrix Business Capability Map Strategy/Capability matrix Capability/Organization matrix Business Model diagram Value Stream catalog Value Stream Stages catalog Value Stream/Capability matrix Value Stream Map

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Line Management (Project Organization); e.g., Project Manager	Operationally achieving on-time, on-budget delivery of a change initiative with an agreed scope.	KEEP INFORMED	Application Communication diagram Organization map Environments and Locations diagram Business Capabilities catalog Capability/Organization matrix Business Capability Map Strategy/Capability matrix Capability/Organization matrix Business Model diagram Value Stream catalog Value Stream Stages catalog Value Stream/Capability matrix Value Stream Map

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Business Process/Functional Expert (Project Organization); e.g., Financials FICO® Functional Consultant, HR Functional Consultant	Adding more detail to the functional requirements of a change initiative based on experience and interaction with business domain experts in the end-user organization.	KEY PLAYERS	<p>Process Flow diagram</p> <p>Business Use-Case diagram</p> <p>Business Service/Information diagram</p> <p>Organization map</p> <p>Application Communication diagram</p> <p>Business Capabilities catalog</p> <p>Capability/Organization matrix</p> <p>Business Capability Map</p> <p>Strategy/Capability matrix</p> <p>Capability/Organization matrix</p> <p>Business Model diagram</p> <p>Value Stream catalog</p> <p>Value Stream Stages catalog</p> <p>Value Stream/Capability matrix</p> <p>Value Stream Map</p>
Product Specialist (Project Organization); e.g., Portal Product Specialist	<p>Specifying technology product designs in order to meet project requirements and comply with the Architecture Vision of the solution.</p> <p>In a packages and packaged services environment, product expertise can be used to identify product capabilities that can be readily leveraged and can provide guidance on strategies for product customization.</p>	KEY PLAYERS	<p>Software Engineering diagram</p> <p>Application/Data matrix</p>

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

Stakeholder	Key Concerns	Class	Catalogs, Matrices, and Diagrams
Suppliers (Outside Services); e.g., Alliance Partners, Key Suppliers	Ensuring that their information exchange requirements are met in order that agreed service contracts with the client organizations can be fulfilled.	KEEP SATISFIED	Business Footprint diagram Business Service/Information diagram Application Communication diagram Business Capabilities catalog Capability/Organization matrix Business Capability Map Strategy/Capability matrix Capability/Organization matrix Business Model diagram Value Stream catalog Value Stream Stages catalog Value Stream/Capability matrix Value Stream Map

Chapter 4: Architecture Patterns

This chapter provides guidelines for using architecture patterns.

4.1 Introduction

Patterns for describing Enterprise Architectures are becoming increasingly important to practitioners. The diverse and multi-disciplinary nature of Enterprise Architecture requires that patterns be developed in different disciplines, domains, and levels of detail.

Previous versions of this standard did not fully embrace architecture patterns due to their perceived lack of maturity. Today, many organizations are using patterns to describe their architectures at various levels ranging from software design patterns to business patterns. It remains true that there is no single standard for describing Enterprise Architecture patterns. However, it can be said that there is a pattern for describing patterns.

4.1.1 Background

A "pattern" has been defined as: "an idea that has been useful in one practical context and will probably be useful in others" (Source: Analysis Patterns — Re-usable Object Models, by M. Fowler).

In the TOGAF Standard, patterns are considered to be a way of putting building blocks into context; for example, to describe a re-usable solution to a problem. Building blocks are what you use: patterns can tell you how you use them, when, why, and what trade-offs you have to make in doing so. Patterns offer the promise of helping the architect to identify combinations of Architecture and/or Solution Building Blocks (ABBs/SBBs) that have been proven to deliver effective solutions in the past, and may provide the basis for effective solutions in the future.

Pattern techniques are generally acknowledged to have been established as a valuable architectural design technique by Christopher Alexander, a buildings architect, who described this approach in his book *The Timeless Way of Building*, published in 1979. This book provides an introduction to the ideas behind the use of patterns, and Alexander followed it with two further books (*A Pattern Language* and *The Oregon Experiment*) in which he expanded on his description of the features and benefits of a patterns approach to architecture.

Software and buildings architects have many similar issues to address, and so it was natural for software architects to take an interest in patterns as an architectural tool. Many papers and books have been published on them since Alexander's 1979 book, perhaps the most renowned being *Design Patterns: Elements of Re-usable Object-Oriented Software* (Gamma et al., 1994). This book describes simple and elegant solutions to specific problems in object-oriented software design.

4.1.2 Content of a Pattern

Several different formats are used in the literature for describing patterns, and no single format has achieved widespread acceptance. However, there is broad agreement on the types of things that a pattern should contain. The headings which follow are taken from *Pattern-Oriented Software Architecture: A System of Patterns* (Buschmann et al., 1996). The elements described below will be found in most patterns, even if different headings are used to describe them.

Name	A meaningful and memorable way to refer to the pattern, typically a single word or short phrase.
Problem	A description of the problem indicating the intent in applying the pattern — the intended goals and objectives to be reached within the context and forces described below (perhaps with some indication of their priorities).
Context	The preconditions under which the pattern is applicable — a description of the initial state before the pattern is applied.
Forces	<p>A description of the relevant forces and constraints, and how they interact/conflict with each other and with the intended goals and objectives. The description should clarify the intricacies of the problem and make explicit the kinds of trade-offs that must be considered. (The need for such trade-offs is typically what makes the problem difficult, and generates the need for the pattern in the first place.) The notion of "forces" equates in many ways to the "qualities" that architects seek to optimize, and the concerns they seek to address, in designing architectures. For example:</p> <ul style="list-style-type: none"> — Security, robustness, reliability, fault-tolerance — Manageability — Efficiency, performance, throughput, bandwidth requirements, space utilization — Scalability (incremental growth on-demand) — Extensibility, evolvability, maintainability — Modularity, independence, re-usability, openness, composability (plug-and-play), portability — Completeness and correctness — Ease-of-construction — Ease-of-use — etc., . . .
Solution	A description, using text and/or graphics, of how to achieve the intended goals and objectives. The description should identify both the solution's static structure and its dynamic behavior — the people and computing actors, and their collaborations. The description may include guidelines for implementing the solution. Variants or specializations of the solution may also be described.
Resulting Context	The post-conditions after the pattern has been applied. Implementing the solution normally requires trade-offs among competing forces.

This element describes which forces have been resolved and how, and which remain unresolved. It may also indicate other patterns that may be applicable in the new context. (A pattern may be one step in accomplishing some larger goal.) Any such other patterns will be described in detail under Related Patterns.

Examples One or more sample applications of the pattern which illustrate each of the other elements: a specific problem, context, and set of forces; how the pattern is applied; and the resulting context.

Rationale An explanation/justification of the pattern as a whole, or of individual components within it, indicating how the pattern actually works, and why — how it resolves the forces to achieve the desired goals and objectives, and why this is "good". The Solution element of a pattern describes the external structure and behavior of the solution: the Rationale provides insight into its internal workings.

Related Patterns

The relationships between this pattern and others. These may be predecessor patterns, whose resulting contexts correspond to the initial context of this one; or successor patterns, whose initial contexts correspond to the resulting context of this one; or alternative patterns, which describe a different solution to the same problem, but under different forces; or co-dependent patterns, which may/must be applied along with this pattern.

Known Uses Known applications of the pattern within existing systems, verifying that the pattern does indeed describe a proven solution to a recurring problem. Known Uses can also serve as Examples.

Patterns may also begin with an Abstract providing an overview of the pattern and indicating the types of problems it addresses. The Abstract may also identify the target audience and what assumptions are made of the reader.

4.1.3 Terminology

Although design patterns have been the focus of widespread interest in the software industry for several years, particularly in the object-oriented and component-based software fields, it is only recently that there has been increasing interest in architecture patterns — extending the principles and concepts of design patterns to the architecture domain.

The technical literature relating to this field is complicated by the fact that many people in the software field use the term "architecture" to refer to software, and many patterns described as "architecture patterns" are high-level software design patterns. This simply makes it all the more important to be precise in the use of terminology.

4.1.3.1 Architecture Patterns and Design Patterns

The term "design pattern" is often used to refer to any pattern which addresses issues of software architecture, design, or programming implementation. In *Pattern-Oriented Software Architecture: A System of Patterns*, the authors define these three types of patterns as follows:

- 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.

These distinctions are useful, but it is important to note that architecture patterns in this context still refers solely to software architecture. Software architecture is certainly an important part of the focus of the TOGAF Standard, but it is not its only focus.

In this section we are concerned with patterns for enterprise system architecting. These are analogous to software architecture and design patterns, and borrow many of their concepts and terminology, but focus on providing re-usable models and methods specifically for the architecting of enterprise information systems — comprising software, hardware, networks, and people — as opposed to purely software systems.

4.1.3.2 *Patterns and the Architecture Continuum*

Although architecture patterns have not (as yet) been integrated into the TOGAF Standard, each of the first four main phases of the ADM (Phases A through D) gives an indication of the stage at which relevant re-usable architecture assets from the Enterprise Architecture Continuum should be considered for use. Architecture patterns are one such asset.

An enterprise that adopts a formal approach to the use and re-use of architecture patterns will normally integrate their use into the Enterprise Architecture Continuum.

4.1.3.3 *Patterns and Views*

Architecture views are selected parts of one or more models representing a complete system architecture, focusing on those aspects that address the concerns of one or more stakeholders. Patterns can provide help in designing such models, and in composing views based on them.

4.1.3.4 *Patterns and Business Scenarios*

Relevant architecture patterns may well be identified in the work on business scenarios.

4.2 Some Pattern Resources

- The Patterns Home Page (refer to hillside.net/patterns) hosted by the Hillside Group provides information about patterns, links to online patterns, papers, and books dealing with patterns, and patterns-related mailing lists
- The Patterns-Discussion FAQ (refer to <http://purl.org/theopengroup/pd-FAQ>) maintained by Doug Lea provides a very thorough and highly readable FAQ about patterns
- *Patterns and Software: Essential Concepts and Terminology* by Brad Appleton (refer to www.bradapp.com/docs/patterns-intro.html) provides another thorough and readable account of the patterns field

- The Service-Oriented Architecture (SOA) Patterns community website (refer to www.soapatterns.org/), dedicated to the ongoing development and expansion of the SOA design pattern catalog
- The Cloud Computing Design Patterns community website (refer to www.cloudpatterns.org)

Chapter 5: Gap Analysis

The technique known as gap analysis is widely used in the TOGAF Architecture Development Method (ADM) to validate an architecture that is being developed. The basic premise is to highlight a shortfall between the Baseline Architecture and the Target Architecture; that is, items that have been deliberately omitted, accidentally left out, or not yet defined.

5.1 Introduction

A key step in validating an architecture is to consider what may have been forgotten. The architecture must support all of the essential information processing needs of the organization. The most critical source of gaps that should be considered is stakeholder concerns that have not been addressed in prior architectural work.

Potential sources of gaps include:

- Business domain gaps:
 - People gaps (e.g., cross-training requirements)
 - Process gaps (e.g., process inefficiencies)
 - Tools gaps (e.g., duplicate or missing tool functionality)
 - Information gaps
 - Measurement gaps
 - Financial gaps
 - Facilities gaps (buildings, office space, etc.)
- Data domain gaps:
 - Data not of sufficient currency
 - Data not located where it is needed
 - Not the data that is needed
 - Data not available when needed
 - Data not created
 - Data not consumed
 - Data relationship gaps

- Applications impacted, eliminated, or created
- Technologies impacted, eliminated, or created

5.2 Suggested Steps

The suggested steps are as follows:

- Draw up a matrix with all the ABBs of the Baseline Architecture on the vertical axis, and all the ABBs of the Target Architecture on the horizontal axis
- Add to the Baseline Architecture axis a final row labeled "New", and to the Target Architecture axis a final column labeled "Eliminated"
- Where an ABB is available in both the Baseline and Target Architectures, record this with "Included" at the intersecting cell
- Where an ABB from the Baseline Architecture is missing in the Target Architecture, each must be reviewed

If it was correctly eliminated, mark it as such in the appropriate "Eliminated" cell. If it was not, an accidental omission in the Target Architecture has been uncovered that must be addressed by reinstating the ABB in the next iteration of the architecture design — mark it as such in the appropriate "Eliminated" cell.

- Where an ABB from the Target Architecture cannot be found in the Baseline Architecture, mark it at the intersection with the "New" row as a gap that needs to be filled, either by developing or procuring the building block

When the exercise is complete, anything under "Eliminated" or "New" is a gap, which should either be explained as correctly eliminated, or marked as to be addressed by reinstating or developing/procuring the building block.

5.3 Example

Figure 5-1 shows an example analysis for ABBs that are services from the Network Services category of the TOGAF Technical Reference Model (TRM), and shows a number of services from the Baseline Architecture missing from the Target Architecture.

Target → Architecture Baseline Architecture ↓	Video Conferencing Services	Enhanced Telephony Services	Mailing List Services	Eliminated Services ↓
Broadcast Services				Intentionally eliminated
Video Conferencing Services	Included			
Enhanced Telephony Services		Potential match		
Shared Screen Services				Unintentionally excluded - a gap in Target Architecture
New →		Gap: Enhanced services to be developed or produced	Gap: To be developed or produced	© The Open Group

Figure 5-1 Gap Analysis Example

Chapter 6: Migration Planning Techniques

This chapter contains a number of techniques used to support migration planning in Phases E and F.

6.1 Implementation Factor Catalog

The technique of creating an Implementation Factor catalog can be used to document factors impacting the architecture Implementation and Migration Plan.

The catalog should include a list of the factors to be considered, their descriptions, and the deductions that indicate the actions or constraints that have to be taken into consideration when formulating the plans.

Factors typically include:

- Risks
- Issues
- Assumptions
- Dependencies
- Actions
- Impacts

An example catalog is shown in [Figure 6-1](#).

Implementation Factor Catalog		
Factor	Description	Deduction
<Name of Factor>	<Description of Factor>	<Impact on Migration Plan>
Change in Technology	Shut down the message centers, saving 700 personnel, and have them replaced by email.	<ul style="list-style-type: none">• Need for personnel training, re-assignment• Email has major personnel savings and should be given priority
Consolidation of Services		
Introduction of New Customer Service		

© The Open Group

Figure 6-1 Implementation Factor Catalog

6.2 Consolidated Gaps, Solutions, & Dependencies Matrix

The technique of creating a Consolidated Gaps, Solutions, and Dependencies matrix allows the architect to group the gaps identified in the domain architecture gap analysis results and assess potential solutions and dependencies to one or more gaps.

This matrix can be used as a planning tool when creating work packages. The identified dependencies will drive the creation of projects and migration planning in Phases E and F.

An example matrix is shown in [Figure 6-2](#).

Consolidated Gaps, Solutions, and Dependencies Matrix				
No.	Architecture	Gap	Potential Solutions	Dependencies
1	Business	New Order Processing Process	Use COTS software tool process Implement custom solution	Drives applications (2)
2	Application	New Order Processing Application	COTS software tool X Develop in-house	
3	Information	Consolidated Customer Information Base	Use COTS customer base Develop customer data mart	

© The Open Group

Figure 6-2 Consolidated Gaps, Solutions, and Dependencies Matrix

6.3 Architecture Definition Increments Table

The technique of creating an Architecture Definition Increments table allows the architect to plan a series of Transition Architectures outlining the status of the Enterprise Architecture at specified times.

A table should be drawn up, as shown in [Figure 6-3](#), listing the projects and then assigning their incremental deliverables across the Transition Architectures.

Architecture Definition - Project Objectives by Increment (Example Only)				
Project	April 2018/2019	April 2019/2020	April 2020/2021	Comments
	Transition Architecture 1: Preparation	Transition Architecture 2: Initial Operational Capability	Transition Architecture 3: Benefits	
Enterprise e-Services Capability	Training and Business Process	e-Licensing Capability	e-Employment Benefits	
IT e-Forms	Design and Build			
IT e-Information Environment	Design and Build Information Environment	Client Common Data Web Content Design and Build	Enterprise Common Data Component Management Design and Build	
...

© The Open Group

Figure 6-3 Architecture Definition Increments Table

6.4 Transition Architecture State Evolution Table

The technique of creating the Transition Architecture State Evolution table allows the architect to show the proposed state of the architectures at various levels using the defined taxonomy (e.g., the TOGAF TRM).

A table should be drawn, listing the services from the taxonomy used in the enterprise, the Transition Architectures, and proposed transformations, as shown in [Figure 6-4](#).

All SBBs should be described with respect to their delivery and impact on these services. They should also be marked to show the progression of the Enterprise Architecture. In the example, where target capability has been reached, this is shown as "new" or "retain"; where capability is transitioned to a new solution, this is marked as "transition"; and where a capability is to be replaced, this is marked as "replace".

Architectural State using the Technical Reference Model				
Sub-Domain	Service	Transition Architecture 1	Transition Architecture 2	Transition Architecture 3
Infrastructure Applications	Information Exchange Services	Solution System A (replace)	Solution System B-1 (transition)	Solution System B-2 (new)
	Data Management Services	Solution System D (retain)	Solution System D (retain)	Solution System D (retain)
...

© The Open Group

Figure 6-4 Transition Architecture State Evolution Table

Another technique (not shown here) is to use color coding in the matrix; for example:

- Green: service SBB in place (either new or retained)
- Yellow: service being transitioned into a new solution
- Red: service to be replaced

6.5 Business Value Assessment Technique

A technique to assess business value is to draw up a matrix based on a value index dimension and a risk index dimension. An example is shown in [Figure 6-5](#). The value index should include criteria such as compliance to principles, financial contribution, strategic alignment, and competitive position. The risk index should include criteria such as size and complexity, technology, organizational capacity, and impact of a failure. Each criterion should be assigned an individual weight.

The index and its criteria and weighting should be developed and approved by senior management. It is important to establish the decision-making criteria before the options are known.

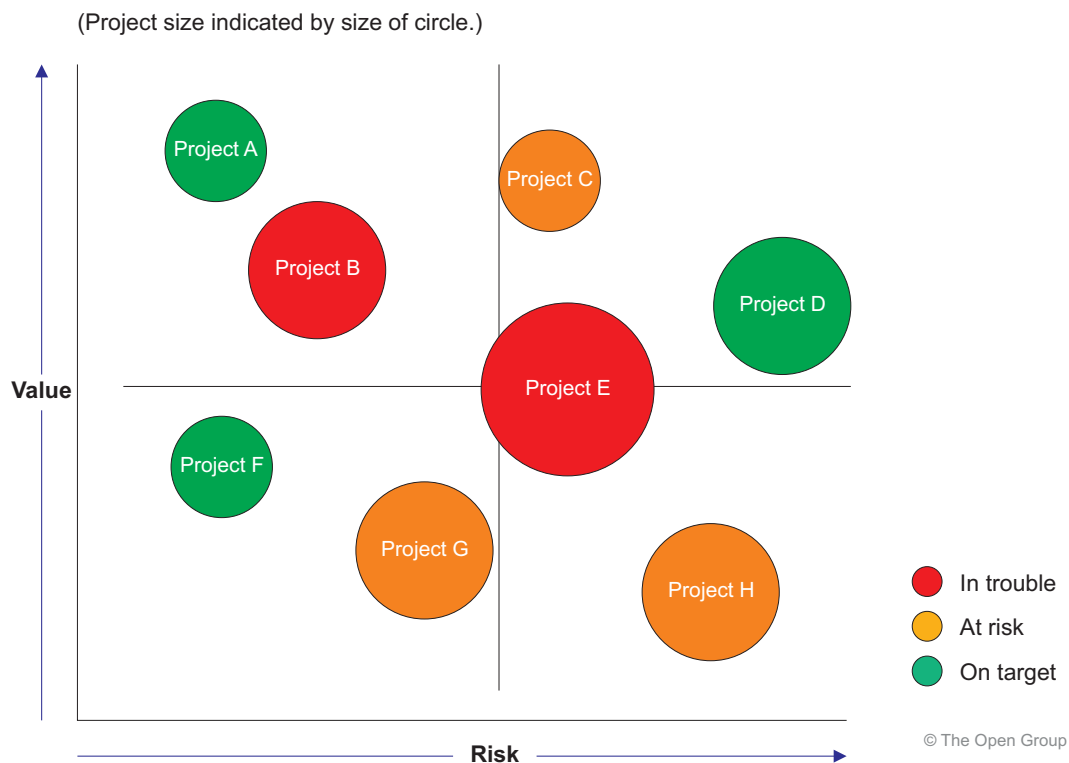


Figure 6-5 Sample Project Assessment with Respect to Business Value and Risk

Chapter 7: Interoperability Requirements

This chapter provides guidelines for defining and establishing interoperability requirements.

7.1 Overview

A definition of interoperability is "the ability to share information and services". Defining the degree to which the information and services are to be shared is a very useful architectural requirement, especially in a complex organization and/or extended enterprise.

The determination of interoperability is present throughout the ADM as follows:

- In the Architecture Vision (Phase A), the nature and security considerations of the information and service exchanges are first revealed within the business scenarios
- In the Business Architecture (Phase B), the information and service exchanges are further defined in business terms
- In the Data Architecture (Phase C), the content of the information exchanges is detailed using the corporate data and/or information exchange model
- In the Application Architecture (Phase C), the way that the various applications are to share the information and services is specified
- In the Technology Architecture (Phase D), the appropriate technical mechanisms to permit the information and service exchanges are specified
- In Opportunities & Solutions (Phase E), the actual solutions (e.g., COTS packages) are selected
- In Migration Planning (Phase F), the interoperability is logically implemented

7.2 Defining Interoperability

There are many ways to define interoperability and the aim is to define one that is consistently applied within the enterprise and extended enterprise. It is best that both the enterprise and the extended enterprise use the same definitions.

Many organizations find it useful to categorize interoperability as follows:

- **Operational or Business Interoperability** defines how business processes are to be shared
- **Information Interoperability** defines how information is to be shared
- **Technical Interoperability** defines how technical services are to be shared or at least connect to one another

From an IT perspective, it is also useful to consider interoperability in a similar vein to Enterprise Application Integration (EAI); specifically:

- **Presentation Integration/Interoperability** is where a common look-and-feel approach through a common portal-like solution guides the user to the underlying functionality of the set of systems
- **Information Integration/Interoperability** is where the corporate information is seamlessly shared between the various corporate applications to achieve, for example, a common set of client information

Normally this is based upon a commonly accepted corporate ontology and shared services for the structure, quality, access, and security/privacy for the information.

- **Application Integration/Interoperability** is where the corporate functionality is integrated and shareable so that the applications are not duplicated (e.g., one change of address service/component; not one for every application) and are seamlessly linked together through functionality such as workflow

This impacts the business and infrastructure applications and is very closely linked to corporate business process unification/interoperability.

- **Technical Integration/Interoperability** includes common methods and shared services for the communication, storage, processing, and access to data primarily in the application platform and communications infrastructure domains

This interoperability is premised upon the degree of rationalization of the corporate IT infrastructure, based upon standards and/or common IT platforms. For example, multiple applications sharing one infrastructure or 10,000 corporate websites using one centralized content management/web server (rather than thousands of servers and webmasters spread throughout the country/globe).

Many organizations create their own interoperability models, such as illustrated in the example below from the Canadian Government. They have a high-level definition of the three classes of interoperability and identify the nature of the information and services that they wish to share. Interoperability is coined in terms of e-enablers for e-Government. Their interoperability breakdown is as follows:

- Information Interoperability:
 - Knowledge management
 - Business intelligence
 - Information management
 - Trusted identity
- Business Interoperability:
 - Delivery networks
 - e-Democracy
 - e-Business
 - Enterprise resource management

- Relationship and case management
- Technical Interoperability:
 - IT infrastructure

In certain architectural approaches, such as system of systems or a federated model, interoperability is a strongly recommended best practice that will determine how the systems interact with each other. A key consideration will be the enterprise's business operating model.

7.3 Enterprise Operating Model

Key to establishing interoperability is the determination of the corporate operating model, where the operating model is "the necessary level of business process integration and standardization for delivering goods and services to customers. An operating model describes how a company wants to thrive and grow. By providing a more stable and actionable view of the company than strategy, the operating model drives the design of the foundation for execution."¹

For example, if lines of business or business units only need to share documents, then the ABBs and SBBs may be simpler than if there is a need to share structured transaction data. Similarly, if the Architecture Vision includes a shared services environment, then it is useful to define the level the services are to be shared.

The corporate operating model will normally indicate what type of interoperability approach will be appropriate. This model should be determined in Phase A (Architecture Vision) if not in Phase B (Business Architecture), and definitely by Phase E (Opportunities & Solutions).

Complex enterprises and/or extended enterprises (e.g., supply chain) may have more than one type of operating model. For example, it is common for the internal operating model (and supporting interoperability model) to differ from the one used for the extended enterprise.

7.4 Refining Interoperability

Implementing interoperability requires the creation, management, acceptance, and enforcement of realistic standards that are SMART (Specific, Measurable, Actionable, Realistic, and Time-bound). Clear measures of interoperability are key to success.

Architecture is the key for identifying standards and facilitated sessions will examine potential pragmatic ways (that fit within the current or emerging business culture) to achieve the requisite degree of interoperability.

Interoperability should be refined so that it meets the needs of the enterprise and/or extended enterprise in an unambiguous way. The refined interoperability measures (degrees, types, and high-level targets) should be part of or referred to the Enterprise Architecture strategic direction.

These measures are instantiated within a transformation strategy that should be embedded within the Target Architecture definition and pragmatically implemented in the Transition Architectures. Upon completion, also update the consolidated gap analysis results and dependencies to ensure that all output from facilitated sessions is captured.

An example of specifying interoperability is the Degrees of Interoperability (used within the

1. *Enterprise Architecture as Strategy* (Ross et al., 2006) provides potential models.

Canadian Department of National Defense and NATO). These organizations were focused on the sharing of information and came up with four degrees of interoperability as follows:

- **Degree 1: Unstructured Data Exchange** involves the exchange of human-interpretable unstructured data, such as the free text found in operational estimates, analysis, and papers
- **Degree 2: Structured Data Exchange** involves the exchange of human-interpretable structured data intended for manual and/or automated handling, but requires manual compilation, receipt, and/or message dispatch
- **Degree 3: Seamless Sharing of Data** involves the automated sharing of data amongst systems based on a common exchange model
- **Degree 4: Seamless Sharing of Information** is an extension of Degree 3 to the universal interpretation of information through data processing based on co-operating applications

These degrees should be further refined and made technically meaningful for each of the degrees. An example refinement of Degree 3 with four subclassifications follows:

- 3A: Formal Message Exchange
- 3B: Common Data Exchange
- 3C: Complete Data Exchange
- 3D: Real-time Data Exchange

The intent is to specify the detailed degrees of interoperability to the requisite level of detail so that they are technically meaningful.

These degrees are very useful for specifying the way that information has to be exchanged between the various systems and provide critical direction to the projects implementing the systems.

Similar measures should be established to determine service/business and technical interoperability.

7.5 Determining Interoperability Requirements

Co-existence between emerging and existing systems, especially during transformation, will be a major challenge and facilitated sessions should attempt to figure out what has to be done to reduce the pain. It is imperative to involve the operations management staff and architects in this step as they will be responsible for operating the portfolio deliverables.

For example, there might be a need for a "wrapper" application (an application that acts as the interface [a.k.a. interpreter] between the legacy application and the emerging infrastructure). Indeed, pragmatically, in the "if it works do not fix it" world, the "wrapper" might become a permanent solution.

Regardless, using the gap analysis results and business scenarios as a foundation, discuss the IT issues and work them through to ensure that all of the gaps are clearly identified and addressed and verify that the organization-specific requirements will be met.

It is important to note that the ensuing development process must include recognition of dependencies and boundaries for functions and should take account of what products are available in the marketplace. An example of how this might be expressed can be seen in the

building blocks example (see the TOGAF Standard — Architecture Content).

If a mechanism such as the Degrees of Interoperability is used, then a matrix showing the interoperability requirements is a useful tool, as illustrated in [Figure 7-1](#) and [Figure 7-2](#), noting that the degree of information sharing is not necessarily symmetrical or bidirectional between systems and/or stakeholders.

The matrix below can be used within the enterprise and/or within the extended enterprise as a way of detailing that information and/or services can be shared. The matrix should start in the Business Architecture (Phase B) to capture the nature of the sharing of information between stakeholders, and evolve to determine what systems share what information in Phase C.

Phase B: Inter-stakeholder Information Interoperability Requirements (Using degrees of information interoperability)							
Stakeholders	A	B	C	D	E	F	G
A		2	3	2	3	3	3
B	2		3	2	3	2	2
C	3	3		2	2	2	3
D	2	2	2		3	3	3
E	4	4	2	3		3	3
F	4	4	2	3	3		2
G	2	2	3	3	3	3	

© The Open Group

Figure 7-1 Business Information Interoperability Matrix

[Figure 7-1](#) shows that Stakeholder A requires structured data exchange (Degree 2) with Stakeholders/Systems B and D, and seamless sharing of data (Degree 3) with Stakeholders/Systems C, E, F, and G.

The business information interoperability matrix should be refined within the Information Systems Architecture using refined measures and specifying the actual systems used by the stakeholders. A sample is shown in [Figure 7-2](#).

Phase C: Inter-system Interoperability Requirements							
	System A	System B	System C	System D	System E	System F	System G
System A		2A	3D	2B	3A	3A	3B
System B	2E		3F	2C	3A	2B	2C
System C	3E	3F		2B	2A	2A	3B
System D	2B	2B	2B		3A	3A	3B
System E	4A	4B	2B	3A		3B	3B
System F	4A	4A	2B	3B	3A		2D
System G	2B	2B	3A	3A	3B	3B	

© The Open Group

Figure 7-2 Information Systems Interoperability Matrix

In [Figure 7-2](#), both the nature of the exchange is more detailed (e.g., Degree 3A *versus* only Degree 3) and the sharing is between specific systems rather than stakeholders. For example, System A shares information with the other systems in accordance with enterprise technical standards.

In many organizations the Business Architectures describe the nature of the information shared between stakeholders and/or organizations (e.g., in defense the term is "operational node"), and the Data Architecture specifies the information shared between systems.

Update the defined target data and Application Architecture (Approved) with the interoperability issues that were raised.

7.6 Reconciling Interoperability Requirements with Potential Solutions

The Enterprise Architect will have to ensure that there are no interoperability conflicts, especially if there is an intention to re-use existing SBBs and/or COTS.

The most significant issue to be addressed is in fact business interoperability. Most SBBs or COTS will have their own business processes embedded. Changing the embedded business processes will often require so much work that the advantages of re-using solutions will be lost. There are numerous examples of this in the past.

Furthermore, there is the workflow aspect between the various systems that has to be taken into account. The Enterprise Architect will have to ensure that any change to the business interoperability requirements is signed off by the Business Architects and architecture sponsors in a revised Statement of Architecture Work.

Chapter 8: Business Transformation Readiness Assessment

This chapter describes a technique known as Business Transformation Readiness Assessment, used for evaluating and quantifying an organization's readiness to undergo change.

This chapter builds on work by the Canadian Government and its Business Transformation Enablement Program (BTEP).

8.1 Introduction

Enterprise Architecture is a major endeavor within an organization and most often an innovative Architecture Vision (Phase A) and supporting Architecture Definition (Phases B to D) will entail considerable change. There are many dimensions to change, but by far the most important is the human element. For example, if the enterprise envisages a consolidation of information holdings and a move to a new paradigm such as service orientation for integrated service delivery, then the human resource implications are major. Potentially coupled with a change-averse culture and a narrowly skilled workforce, the most sound and innovative architecture could go nowhere.

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 transformation in Phases E and F. This will be a joint effort between corporate (especially human resources) staff, lines of business, and IT planners.

The recommended activities in an assessment of an organization's readiness to address business transformation are:

- Determine the readiness factors that will impact the organization
- Present the readiness factors using maturity models
- Assess the readiness factors, including determination of readiness factor ratings
- Assess the risks for each readiness factor and identify improvement actions to mitigate the risk
- Work these actions into Phase E and F Implementation and Migration Plan

8.1.1 Business Transformation Enablement Program (BTEP)

The Canadian Government Business Transformation Enablement Program (BTEP) provides guidance on how to identify the business transformation-related issues.

The BTEP recommends that all projects conduct a transformation readiness assessment to at least uncover the business transformation issues. This assessment is based upon the determination and analysis/rating of a series of readiness factors. The outcome is a deeper understanding of the challenges and opportunities that could be presented in the course of the endeavor. Many of the challenges translate directly into risks that have to be addressed, monitored, and, if possible, mitigated.

The following sections describe Business Transformation Readiness Assessment using the BTEP method, including some lessons learned. Readers should keep in mind that most organizations will have their own unique set of factors and criteria, but most are similar.

8.2 Determine Readiness Factors

The first step is to determine what factors will impact on the business transformation associated with the migration from the Baseline to Target Architectures.

This can be best achieved through the conduct of a facilitated workshop with individuals from different parts of the organization. It is important that all perspectives are sought as the issues will be varied. In this workshop it is very useful to start off with a tentative list of factors that participants can re-use, reject, augment, or replace.

An example set of factors drawn from the BTEP follows:

- **Vision** is the ability to clearly define and communicate what is to be achieved

This is where management is able to clearly define the objectives, in both strategic and specific terms. Leadership in defining vision and needs comes from the business side with IT input. Predictable and proven processes exist for moving from vision to statement of requirements. The primary drivers for the initiative are clear. The scope and approach of the transformation initiative have been clearly defined throughout the organization.

- **Desire, Willingness, and Resolve** is the presence of a desire to achieve the results, willingness to accept the impact of doing the work, and the resolve to follow through and complete the endeavor

There is active discussion regarding the impact that executing the project may have on the organization, with a clear indication of the intent to accept the impacts. Key resources (e.g., financial, human, etc.) are allocated for the endeavor and top executives project the clear message that the organization will follow through; a message that identifies the effort as well as the benefits. Organizationally there is a history of finishing what is started and of coming to closure on issues in the timeframes needed and there is agreement throughout the organization that the transformation initiative is the "right" thing to do.

- **Need**, in that there is a compelling need to execute the endeavor

There are clear statements regarding what the organization will not be able to do if the project does not proceed, and equally clear statements of what the project will enable the organization to do. There are visible and broadly understood consequences of endeavor failure and success criteria have been clearly identified and communicated.

- **Business Case** exists that creates a strong focus for the project, identifying benefits that must be achieved and thereby creating an imperative to succeed

The business case document identifies concrete benefits (revenues or savings) that the organization is committed to deliver and clearly and unquestionably points to goals that the organization is committed to achieving.

- **Funding**, in the form of a clear source of fiscal resources, exists that meets the endeavor's potential expenditures
- **Sponsorship and Leadership** exists and is broadly shared, but not so broad as to diffuse accountability

Leadership keeps everyone "on board" and keeps all focused on the strategic goals. The endeavor is sponsored by an executive who is appropriately aligned to provide the leadership the endeavor needs and able to articulate and defend the needs of the endeavor at the senior management level. These executive sponsors are and will remain engaged throughout.

- **Governance** is the ability to engage the involvement and support of all parties with an interest in or responsibility to the endeavor with the objective of ensuring that the corporate interests are served and the objectives achieved

There are clearly identified stakeholders and a clear sense of their interest in and responsibility to the project; a culture that encourages participation towards corporate rather than local objectives; a history of being able to successfully manage activities that cross interest areas; a culture that fosters meaningful, as opposed to symbolic, participation in management processes; and a commitment to ongoing project review and challenge and openness to outside advice.

- **Accountability** is the assignment of specific and appropriate responsibility, recognition of measurable expectations by all concerned parties, and alignment of decision-making with areas of responsibility and with where the impact of the decisions will be felt

Accountability is aligned with the area where the benefits of success or consequences of failure of the endeavor will be felt as well as with the responsibility areas.

- **Workable Approach and Execution Model** is an approach that makes sense relative to the task, with a supporting environment, modeled after a proven approach

There are clear notions of the client and the client's role relative to the builder or prime contractor and the organization is experienced with endeavors of this type so that the processes, disciplines, expertise, and governance are already in place, proven, and available to apply to the transformation endeavor. All the players know their roles because they have played them before with success. In particular, the roles of "client" and "systems builder" are mature and stable. There is a communication plan covering all levels of the organization and meeting the needs ranging from awareness to availability of technical detail. There is a reward and recognition plan in place to recognize teams and individuals who use good change management practices, planning and prevention of crisis behaviors, and who reinforce behaviors appropriate to the new way of doing business. It is clear to everyone how implementation will occur, how it will be monitored, and how realignment actions will be made and there are adequate resources dedicated for the life of the transformation.

- **IT Capacity to Execute** is the ability to perform all the IT tasks required by the project, including the skills, tools, processes, and management capability

There has been a recent successful execution of a similar endeavor of similar size and complexity and there exist appropriate processes, discipline, skills, and a rationale model for deciding what skills and activities to source externally.

- **Enterprise Capacity to Execute** is the ability of the enterprise to perform all the tasks required by the endeavor, in areas outside of IT, including the ability to make decisions within the tight time constraints typical to project environments based upon the recent successful execution of a similar endeavor of at least half the size and complexity

There exist non-IT-specific processes, discipline, and skills to deal with this type of endeavor. The enterprise has a demonstrated ability to deal with the type of ongoing project/portfolio management issues and requirements. There is a recognition of the need for knowledge and skill-building for the new way of working as well as the value of a formal gap analysis for skills and behavior.

- **Enterprise Ability to Implement and Operate** the transformation elements and their related business processes, absorb the changes arising from implementation, and ongoing ability to operate in the new environment

The enterprise has a recent proven ability to deal with the change management issues arising from new processes and systems and has in place a solid disciplined and process-driven service management program that provides operations, maintenance, and support for existing systems.

Once the factors have been identified and defined, it is useful to call a follow-on workshop where the factors shall be assessed in some detail in terms of their impact/risk. The next section will deal with preparing for an effective assessment of these factors.

8.3 Present Readiness Factors

Once the factors are determined, it is necessary to present them in such a way that the assessment is clear and the maximum value is derived from the participants.

One such presentation is through the use of maturity models. If each factor is converted into a maturity model (a re-usable governance asset as well) accompanied by a standard worksheet template containing all of the information and deductions that have to be gathered, it can be a very useful tool.

The maturity model should enable participants to:

- Assess their current (Baseline Architecture) maturity level
- Determine the target maturity level that would have to be achieved to realize the Target Architecture
- Determine an intermediate target that would be achievable in a lesser timeframe

The care spent preparing the models (which is not insignificant) will be recouped by a focused workshop that will rapidly go through a significant number of factors.

It is important that each factor be well-defined and that the scope of the Enterprise Architecture endeavor (preliminary planning) be reflected in the models to keep the workshop participants focused and productive.

Circulating the models before the workshop for comments would be useful, if only to ensure that they are complete as well as allowing the participants to prepare for the workshop. Note that the model shown below also has a recommended target state put in by the Enterprise Architect; this again acts as governance.

An example of a maturity model is shown in [Figure 8-1](#) for one of the BTEP factors.

Business Transformation Readiness Assessment - Maturity Model					
Factor 2: Need for Enterprise Information Architecture			Class	Organizational Context	
			BTEP Readiness Factor	YES	
Definition	There is recognition by the organization that information is a strategic corporate asset requiring stewardship. There is also recognition that the data is not universally understandable, of requisite quality, and accessible.				
Maturity Model Levels					
0 Not defined	1 <i>Ad Hoc</i>	2 Repeatable	3 Defined	4 Managed	5 Optimized
Information is not recognized as an asset. There is no clear stewardship of data.	Data Management (DM) concepts are intuitively understood and practiced on an <i>ad hoc</i> basis. Stewardship of the data is informal. Data is recognized by certain internal experts and senior management as being of strategic importance to the organization. Focus is primarily on technically managing redundant data at the applications level.	Many parts of the organization value information/data as a strategic asset. Internal DM experts maintain clear lines of responsibility and stewardship of the data, organized along lines of business and at all senior levels. Staff put into practice DM principles and standards in their daily activities.	Data is recognized as a strategic asset in most parts of the organization, and throughout most levels from operations to senior management. Resources are committed to ensuring strong stewardship of data at the lower management and information expert levels.	Data is recognized as a strategic asset in all parts of the organization, and throughout most levels from operations to senior management. Resources are committed to ensuring strong stewardship of data at the senior management and information expert levels.	Data is treated in all levels throughout the organization as a strategic asset to be exploited and re-used. Data products and services are strongly integrated with the management practice of the organization. All staff are empowered and equipped to take stewardship of information, and are seen as "knowledge workers".
				Recommended Target State	

© The Open Group

Figure 8-1 Business Transformation Readiness Assessment — Maturity Model

8.4 Assess Readiness Factors

Ideally, the factors should be assessed in a multi-disciplinary workshop. Using a mechanism such as maturity models, Enterprise Architects will normally have to cover a great deal of ground in little time.

The use of a series of templates for each factor would expedite the assessment, and ensure consistency across the wide range of factors.

The assessment should address three things, namely:

- Readiness Factor Vision
- Readiness Factor Rating
- Readiness Factor Risks & Actions

8.4.1 Readiness Factor Vision

The vision for a readiness factor is the determination of where the enterprise has to evolve to address the factor. First, the factor should be assessed with respect to its base state and then its target state.

For example, if the "IT capacity to execute" factor is rated as low, the factor should ideally be at "high" to realize the Target Architecture Vision. An intermediate target might be useful to direct the implementation. Maturity models are excellent vehicles to guide this determination.

8.4.2 Readiness Factor Rating

Once the factor visions are established, then it is useful to determine how important each factor is to the achievement of the Target Architecture as well as how challenging it will be to migrate the factor into an acceptable visionary state.

The BTEP uses a Readiness Rating Scheme that can be used as a start point for any organization in any vertical. Each one of the readiness factors are rated with respect to:

- **Urgency**, whereby if a readiness factor is urgent, it means that action is needed before a transformation initiative can begin
- **Readiness Status**, which is rated as either Low (needs substantial work before proceeding), Fair (needs some work before proceeding), Acceptable (some readiness issues exist; no showstoppers), Good (relatively minor issues exist), or High (no readiness issues)
- **Degree of Difficulty to Fix** rates the effort required to overcome any issues identified as either No Action Needed, Easy, Moderate, or Difficult

Although a more extensive template can be used in the workshop, it is useful to create a summary table of the findings to consolidate the factors and provide a management overview. A summary is shown in [Figure 8-2](#).

Business Factor Assessment Summary				
Ser	Readiness Factor	Urgency	Readiness Status	Degree of Difficulty to Fix
1	Vision			
2	Desire/willingness/resolve			
3	Need			
4	Business case			
5	Funding			
6	Sponsorship and leadership			
7	Governance			
8	Accountability			
9	Workable approach and execution model			
10	IT capacity to execute			
11	Departmental capacity to execute			
12	Ability to implement and operate			

© The Open Group

Figure 8-2 Summary Table of Business Transformation Readiness Assessment

8.4.3 Readiness Factor Risks & Actions

Once the factors have been rated and assessed, derive a series of actions that will enable the factors to change to a favorable state.

Each factor should be assessed with respect to risk using the process highlighted in [Chapter 9](#), including an estimate of impact and frequency.

Each factor should be discretely assessed and a series of improvement actions outlined. Before starting anew, existing actions outlined in the architectures should be checked first before creating new ones.

These newly identified actions should then be formally incorporated into the emerging Implementation and Migration Plan.

From a risk perspective, these actions are designed to mitigate the risks and produce an acceptable residual risk. As risks, they should be part of the risk management process and closely monitored as the Enterprise Architecture is being implemented.

8.5 Readiness and Migration Planning

The assessment exercise will provide a realistic assessment of the organization and will be a key input into the strategic migration planning that will be initiated in Phase E and completed in Phase F. It is important to note whether the business transformation actions will be on the vision's critical path and, if so, determine how they will impact implementation. There is no point deploying new IT capability without employees trained to use it and support staff ready to sustain it.

The readiness factors, as part of an overall Implementation and Migration Plan, will have to be continuously monitored (Phase G) and rapid corrective actions taken through the IT governance framework to ensure that the defined architectures can be implemented.

The readiness factors assessment will be a living document and during the migration planning and execution of the Transition Architectures, the business transformation activities will play a key role.

8.6 Marketing the Implementation Plan

The Architecture Definition should not be widely circulated until the business transformation issues are identified and mitigated, and the associated actions part of an overall "marketing" plan for the vision and the Implementation and Migration Plan.

For example, the consolidation of information holdings could result in hundreds of lost jobs and this vision should not be announced before a supporting business transformation/human resources plan is formulated to retrain or support the workers' quest for new employment.

The business transformation workshops are a critical part of the Communications Plan whereby key individuals from within the organization gather to assess the implications of transforming the enterprise. To do this they will become aware of the Architecture Vision and architecture definition (if they were not already involved through the business scenarios and Business Architecture). This group will feel ownership of the Enterprise Architecture, recognizing the Enterprise Architect as a valuable steward.

Their determination of the factors will again create a culture of understanding across the enterprise and provide useful insights for the Implementation and Migration Plan.

The latter plan should include a Communications Plan, especially to keep the affected personnel informed. In many cases collaborating with the unions and shop stewards will further assist a humane (and peaceful) transition to the target state.

8.7 Conclusion

In short, Enterprise Architecture implementation will require a deep knowledge and awareness of all of the business transformation factors that impact transitioning to the visionary state. With the evolution of IT, the actual technology is not the real issue any more in Enterprise Architecture, but the critical factors are most often the cultural ones. Any Implementation and Migration Plan has to take both into consideration. Neglecting these and focusing on the technical aspects will invariably result in an implementation that falls short of realizing the real promise of a visionary Enterprise Architecture.

Chapter 9: Risk Management

This chapter describes risk management, which is a technique used to mitigate risk when implementing an architecture project.

9.1 Introduction

There will always be risk with any architecture/business transformation effort. It is important to identify, classify, and mitigate these risks before starting so that they can be tracked throughout the transformation effort.

Mitigation is an ongoing effort and often the risk triggers may be outside the scope of the transformation planners (e.g., merger, acquisition) so planners must monitor the transformation context constantly.

It is also important to note that the Enterprise Architect may identify the risks and mitigate certain ones, but it is within the governance framework that risks have to be first accepted and then managed.

There are two levels of risk that should be considered, namely:

1. **Initial Level of Risk:** risk categorization prior to determining and implementing mitigating actions
2. **Residual Level of Risk:** risk categorization after implementation of mitigating actions (if any)

The process for risk management is described in the following sections and consists of the following activities:

- Risk classification
- Risk identification
- Initial risk assessment
- Risk mitigation and residual risk assessment
- Risk monitoring

9.2 Risk Classification

Risk is pervasive in any Enterprise Architecture activity and is present in all phases within the ADM. From a management perspective, it is useful to classify the risks so that the mitigation of the risks can be executed as expeditiously as possible.

One common way for risks to be classified is with respect to impact on the organization (as discussed in [Section 9.4](#)), whereby risks with certain impacts have to be addressed by certain levels of governance.

Risks are normally classified as time (schedule), cost (budget), and scope but they could also include client transformation relationship risks, contractual risks, technological risks, scope and complexity risks, environmental (corporate) risks, personnel risks, and client acceptance risks.

Another way of delegating risk management is to further classify risks by architecture domains. Classifying risks as business, information, applications, and technology is useful but there may be organizationally-specific ways of expressing risk that the corporate Enterprise Architecture directorate should adopt or extend rather than modify.

Ultimately, Enterprise Architecture risks are corporate risks and should be classified and as appropriate managed in the same or extended way.

9.3 Risk Identification

The maturity and transformation readiness assessments will generate a great many risks. Identify the risks and then determine the strategy to address them throughout the transformation.

The use of Capability Maturity Models (CMMs) is suitable for specific factors associated with architecture delivery to first identify baseline and target states and then identify the actions required to move to the target state. The implications of *not* achieving the target state can result in the discovery of risks. Refer to [Chapter 8](#) for specific details.

Risk documentation is completed in the context of a Risk Management Plan, for which templates exist in standard project management methodologies — e.g., Project Management Book of Knowledge (PMBOK®) and PRINCE2® — as well as with the various government methodologies.

Normally these methodologies involve procedures for contingency planning, tracking and evaluating levels of risk, reacting to changing risk level factors, as well as processes for documenting, reporting, and communicating risks to stakeholders.

9.4 Initial Risk Assessment

The next step is to classify risks with respect to effect and frequency in accordance with scales used within the organization. Combine effect and frequency to come up with a preliminary risk assessment.

There are no hard and fast rules with respect to measuring effect and frequency. The following guidelines are based upon existing risk management best practices.

Effect could be assessed using the following example criteria:

- **Catastrophic** infers critical financial loss that could result in bankruptcy of the organization
- **Critical** infers serious financial loss in more than one line of business leading to a loss in productivity and no return on investment on the IT investment
- **Marginal** infers a minor financial loss in a line of business and a reduced return on investment on the IT investment
- **Negligible** infers a minimal impact on a line of business' ability to deliver services and/or products

Frequency could be indicated as follows:

- **Frequent**: likely to occur very often and/or continuously
- **Likely**: occurs several times over the course of a transformation cycle
- **Occasional**: occurs sporadically
- **Seldom**: remotely possible and would probably occur not more than once in the course of a transformation cycle
- **Unlikely**: will probably not occur during the course of a transformation cycle

Combining the two factors to infer impact would be conducted using a heuristically-based but consistent classification scheme for the risks. A potential scheme to assess corporate impact could be as follows:

- **Extremely High Risk (E)**: the transformation effort will most likely fail with severe consequences
- **High Risk (H)**: significant failure of parts of the transformation effort resulting in certain goals not being achieved
- **Moderate Risk (M)**: noticeable failure of parts of the transformation effort threatening the success of certain goals
- **Low Risk (L)**: certain goals will not be wholly successful

These impacts can be derived using a classification scheme, as shown in [Figure 9-1](#).

Corporate Risk Impact Assessment					
Effect	Frequency				
	Frequent	Likely	Occasional	Seldom	Unlikely
Catastrophic	E	E	H	H	M
Critical	E	H	H	M	L
Marginal	H	M	M	L	L
Negligible	M	L	L	L	L

© The Open Group

Figure 9-1 Risk Classification Scheme

9.5 Risk Mitigation and Residual Risk Assessment

Risk mitigation refers to the identification, planning, and conduct of actions that will reduce the risk to an acceptable level.

The mitigation effort could be a simple monitoring and/or acceptance of the risk to a full-blown contingency plan calling for complete redundancy in a Business Continuity Plan (with all of the associated scope, cost, and time implications).

Due to the implications of this risk assessment, it has to be conducted in a pragmatic but systematic manner. With priority going to frequent high impact risks, each risk has to be mitigated in turn.

9.6 Conduct Residual Risk Assessment

Once the mitigation effort has been identified for each one of the risks, re-assess the effect and frequency and then recalculate the impacts and see whether the mitigation effort has really made an acceptable difference. The mitigation efforts will often be resource-intensive and a major outlay for little or no residual risk should be challenged.

Once the initial risk is mitigated, then the risk that remains is called the "residual risk". The key consideration is that the mitigating effort actually reduces the corporate impact and does not just move the risk to another similarly high quadrant. For example, changing the risk from frequent/catastrophic to frequent/critical still delivers an extremely high risk. If this occurs, then the mitigation effort has to be re-considered.

The final deliverable should be a transformation risk assessment that could be structured as a worksheet, as shown in [Figure 9-2](#).

Risk ID	Risk	Preliminary Risk			Mitigation	Residual Risk		
		Effect	Frequency	Impact		Effect	Frequency	Impact

© The Open Group

Figure 9-2 Sample Risk Identification and Mitigation Assessment Worksheet

9.7 Risk Monitoring and Governance (Phase G)

The residual risks have to be approved by the IT governance framework and potentially in corporate governance where business acceptance of the residual risks is required.

Once the residual risks have been accepted, then the execution of the mitigating actions has to be carefully monitored to ensure that the enterprise is dealing with residual rather than initial risk. The risk identification and mitigation assessment worksheets are maintained as governance artifacts and are kept up-to-date in Phase G (Implementation Governance) where risk monitoring is conducted.

Implementation governance can identify critical risks that are not being mitigated and might require another full or partial ADM cycle.

9.8 Summary

Risk management is an integral part of Enterprise Architecture. Practitioners are encouraged to use their corporate risk management methodology or extend it using the guidance in this chapter. In the absence of a formal corporate methodology, architects can use the guidance in this chapter as a best practice.

Chapter 10: Architecture Alternatives and Trade-Offs

This chapter describes a technique to identify alternative Target Architectures and perform trade-offs between the alternatives.

10.1 Concept

There is often more than one possible Target Architecture that would conform to the Architecture Vision, Architecture Principles, and Requirements. It is important to identify alternative Target Architectures and build understanding of different possibilities and identify trade-offs between the alternatives. Creating an architecture normally requires trade-offs among competing forces. Presenting different alternatives and trade-offs to stakeholders helps architects to extract hidden agendas, principles, and requirements that could impact the final Target Architecture.

10.2 Method

It is most common that a single alternative does not exist that will meet all stakeholders' concerns. The TOGAF Standard supports a technique to investigate different alternatives and to discuss these with the stakeholders. Commonly, alternatives are defined per domain. This is done to simplify the analysis of the different alternatives. Of course, the alternatives per domain can be merged into an overall analysis of the alternatives for the whole architecture.

Figure 10-1 illustrates the architecture trade-off method.

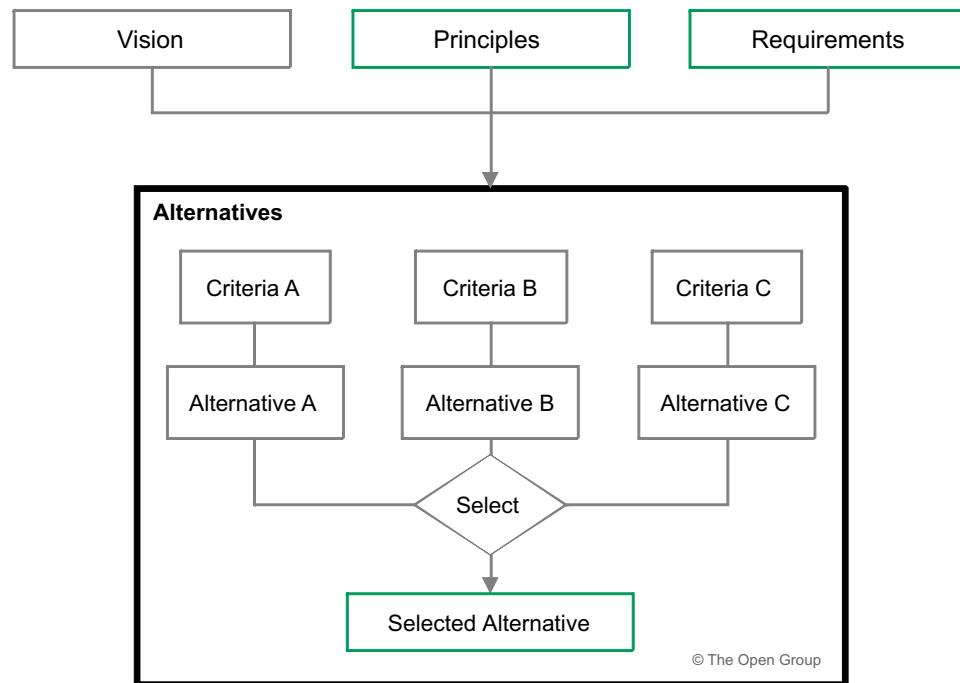


Figure 10-1 Architecture Trade-Off Method

The first part of the method uses the vision, principles, requirements, and other information to select sets of criteria fitting for different alternatives.

This second part of the method defines alternatives based on the criteria and builds understanding of each.

The third part of this method will either select one of the alternatives, or else combine features from more than one, to create the proposed alternative. Perform the following activities in just enough detail to support that decision. The method can be used for any phase at any level of an architecture.

10.2.1 Criteria

The criteria are used for the different alternatives and are derived from many different inputs to the architecture. Consider the influence of architecture principles, requirements, vision, and stakeholder concerns.

Each alternative will have distinct advantages or disadvantages that will need to be discussed and agreed with stakeholders. Additional viewpoints and views may be needed to allow stakeholders to explore the alternatives and understand any dependencies, risks, and uncertainties.

Typical examples of alternative types (based on criteria) include:

- Flexible alternative
- Time and cost of realizing the alternative, including any transitions and plateaus ("islands of stability")
- Time period over which estimated benefits of the alternative will be achieved
- Adherence to architecture styles or guidelines
- Solution delivery method (e.g., re-use, develop, buy)
- Minimal impact on business capabilities during implementation of the alternative
- Minimized risk associated with the alternative and any mitigating actions needed
- etc., . . .

10.2.2 Identify Alternatives

Identify a set of possible alternatives using the Architecture Vision, Principles, and Requirements.

For each alternative:

1. Define the overview criteria for the alternative
Use the Architecture Vision, Principles, and Requirements to define the criteria for the alternative. The criteria can be applied at different abstraction levels and ADM phases to identify different architecture alternatives.
2. Describe the architecture for the alternative
Create a set of necessary architecture views to reach a proper understanding of the impact of the alternative. Add any other needed information. Do not go into too much detail. It is important, however, to carry out a good impact assessment and identify interdependencies between alternatives and the existent landscape and have a complete picture of the implications of the alternative implementation.
3. Estimate gaps between the baseline and this alternative
Based on current understanding of the baseline state, outline the gaps that exist between the baseline and this alternative. If the baseline has not yet been defined, this gap analysis will be informal. More detail on how to do gap analysis appears in [Chapter 5](#).
4. Understand the impacts and trade-offs of the alternative across the Architecture Landscape:
 - Identify the impact that the alternative will have on any existing architectures, and on any Transition Architecture within the Architecture Landscape
 - Identify the impact the alternative will have on any running or planned implementation projects
 - Identify constraints imposed on this alternative by any running or planned implementation projects

- Identify impacts on architecture in other ADM phases in this architecture project
- Identify Architecture Requirements/Change Requests from this architecture that will constrain other architectures
- Identify the final value delivered by the alternative, to what extent it covers the gap to reach the future state, and the purpose of the iteration

10.2.3 Choose from Alternatives and Define in Detail

This step draws on the alternatives to select or define an alternate alternative. Use trade-off analysis to resolve conflicts between alternatives:

1. Understand the strengths and weaknesses of each alternative
2. Compare the alternatives based on how well they align with the defined criteria
3. Select the most suitable alternative or combine features from more than one of the alternatives, to define an alternate alternative in collaboration with stakeholders
4. Assemble the alternative:
 - Finalize the description of the alternative
 - Ensure that all the architecture viewpoints identified have been worked through for the alternative
 - Ensure that the alternative is defined in enough detail to support decision-making
5. Resolve impacts across the Architecture Landscape
6. Conduct formal stakeholder review to determine alternative decision and funding

The impact analysis of the alternatives was done in just enough detail to choose between them.

Index

ABB.....	39
application principles	
examples	16
architecture alternatives.....	73
Architecture Continuum	42
architecture pattern.....	39, 41
content	40
Architecture Principles	3
characteristics	4
components.....	4
criteria	5
architecture view	22, 42
architecture viewpoint	22
BTEP	59-60
Readiness Rating Scheme.....	64
Business Continuity Plan	70
business principles	
examples	7
Business Transformation Readiness Assessment.....	59
concern	22
COTS	16
data principles	
examples	11
Degrees of Interoperability.....	56
design pattern	41
enterprise operating model.....	55
enterprise principles	3
gap analysis	45
GOTS.....	16
governance	70
Hillside Group	42
idiom	42
initial risk assessment.....	68
interoperability	
categories.....	53
refining	55
requirements	53
Java	16
maturity model	62
Migration Planning	66
pattern resources.....	42
patterns home page.....	42
patterns-discussion FAQ.....	42
readiness factors.....	60

readiness planning.....	66
residual risk assessment	70
risk assessment	68
risk classification	68
risk identification	68
risk management	67
best practice	69
Risk Management Plan.....	68
risk monitoring	70
SANS Institute	11
SBB	39
SMART	55
stakeholder	22
stakeholder management	21
process steps	22
steward	13
technology principles	
examples	17
transformation readiness assessment	60
TRM	46
trustee.....	13

The Open Group Standard

TOGAF® Standard — Applying the ADM

The Open Group

Copyright © 2005-2022, The Open Group

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

Any use of this publication for commercial purposes is subject to the terms of the Annual Commercial License relating to it. For further information, see www.opengroup.org/legal/licensing.

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

Chapter	1	Introduction	1
	1.1	Using the TOGAF Framework with Different Architecture Styles.....	1
Chapter	2	Applying Iteration to the ADM	3
	2.1	Overview.....	3
	2.2	Iteration Cycles.....	4
	2.3	Classes of Architecture Engagement	5
	2.4	Approaches to Architecture Development	9
	2.5	Iteration Considerations.....	10
	2.5.1	Iteration between ADM Cycles	10
	2.5.2	Iteration within an ADM Cycle	12
	2.6	Conclusions	15
Chapter	3	Applying the ADM Across the Architecture Landscape	17
	3.1	Overview.....	17
	3.2	Architecture Landscape.....	17
	3.3	Developing Architectures at Different Levels	19
	3.4	Organizing the Architecture Landscape to Understand the State of the Enterprise.....	19
Chapter	4	Architecture Partitioning	21
	4.1	Overview.....	21
	4.2	Applying Classification to Create Partitioned Architectures.....	21
	4.2.1	Activities within the Preliminary Phase	23
	4.3	Integration.....	24
		Index	27
List of Figures			
	2-1	Iteration Cycles.....	4
	2-2	Classes of Enterprise Architecture Engagement.....	5
	2-3	A Hierarchy of ADM Processes Example	10
	2-4	Activity by Iteration for Baseline First Architecture Definition.....	12
	2-5	Activity by Iteration for Target First Architecture Definition.....	12
	3-1	Summary Classification Model for Architecture Landscapes	17

3-2	Summary of Architecture Continuum.....	18
4-1	Allocation of Teams to Architecture Scope	24
4-2	Architecture Content Aggregation	25

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 Frameworks
- 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](#).

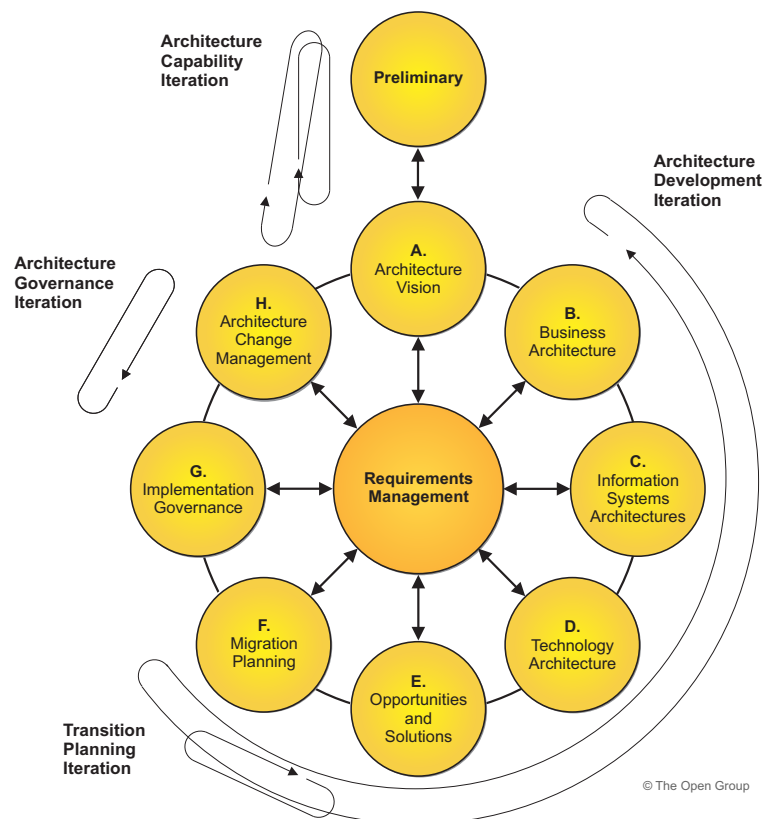


Figure 2-1 Iteration Cycles

- **Architecture Capability** iterations support the creation¹ 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.

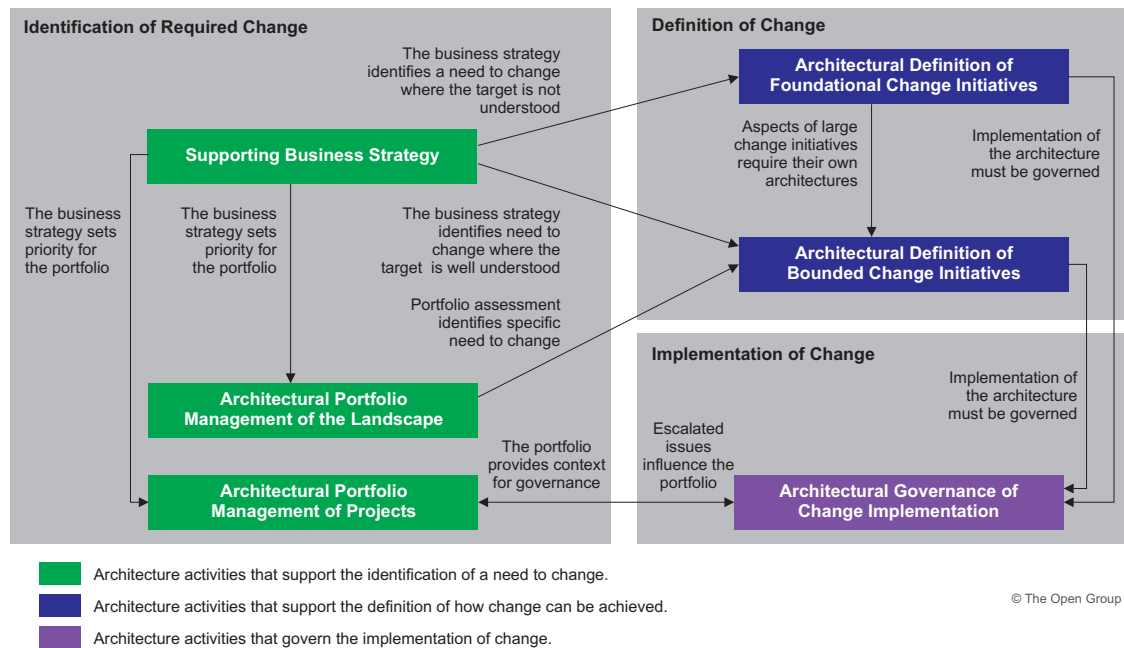


Figure 2-2 Classes of Enterprise Architecture Engagement

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	<p>As the business strategies, objectives, goals, and drivers change, it is necessary for the enterprise to change in order to maintain alignment.</p> <p>The creation of new business strategies can be supported by Enterprise Architecture by:</p> <ul style="list-style-type: none"> ■ 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	<p>It is common practice across large organizations for a service management organization to provide operational reporting and management of the IT portfolio.</p> <p>Enterprise Architecture can add a further dimension to service management reporting, by supporting a linkage between operational performance and the strategic need for IT.</p> <p>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.</p>
	Architectural Portfolio Management of Projects	<p>It is common practice across large organizations for a program management organization to provide operational reporting and management of the change portfolio.</p> <p>Enterprise Architecture can add a further dimension to project portfolio management reporting, by supporting a linkage between project scope, architectural impact, and business value.</p> <p>Architectural factors can be added to other quantitative project factors to support strategic decision-making on project priority and funding levels.</p>
Definition of Change	Architectural Definition of Foundational Change Initiatives	<p>Foundational change initiatives are change efforts that have a known objective, but are not strictly scoped or bounded by a shared vision or requirements.</p> <p>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.</p> <p>Once the problem is more effectively understood, it is possible to define appropriate solutions and to align stakeholders around a common vision and purpose.</p>

Area of Engagement	Architecture Engagement	Description
	Architectural Definition of Bounded Change Initiatives	<p>Bounded change initiatives are change efforts that typically arise as the outcome of a prior architectural strategy, evaluation, or vision.</p> <p>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.</p>
Implementation of Change	Architectural Governance of Change Implementation	<p>Once an architectural solution model has been defined, it provides a basis for design and implementation.</p> <p>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.</p>

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.

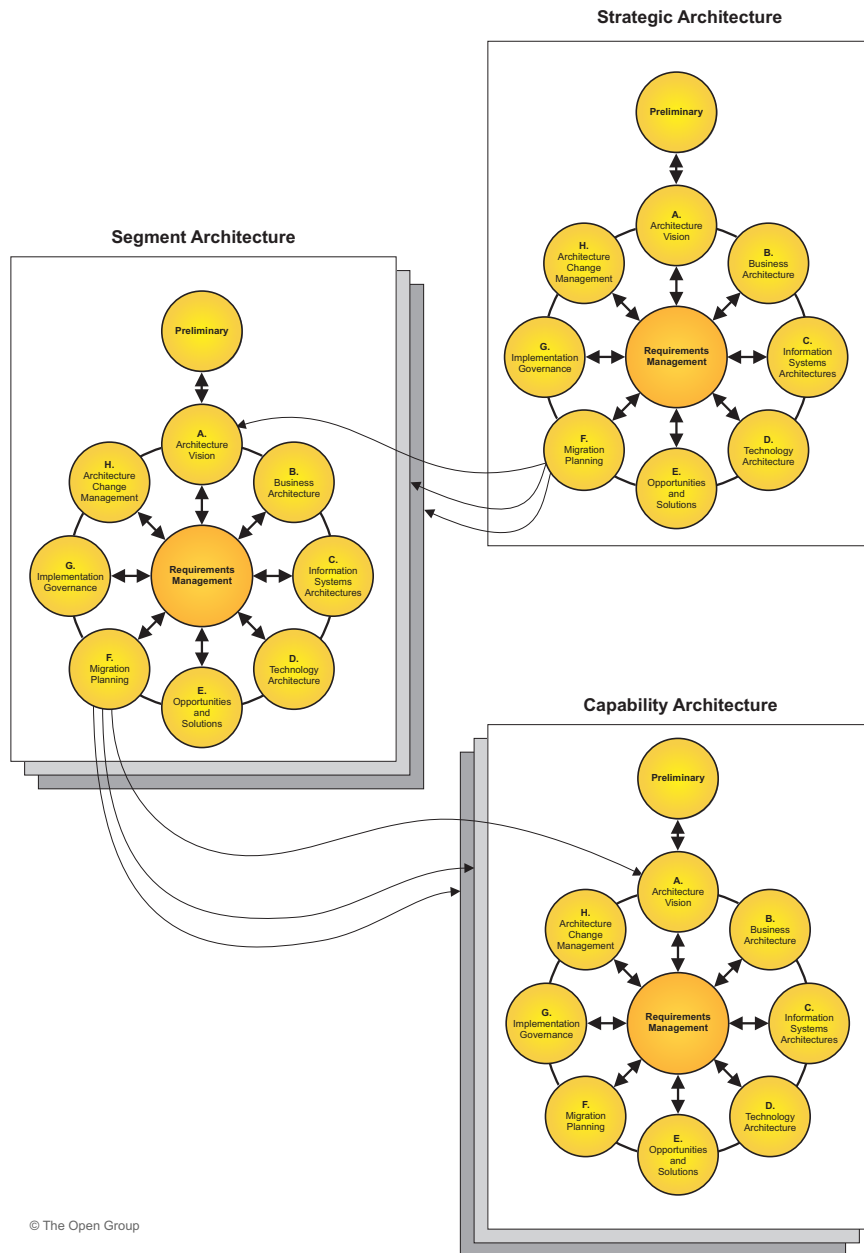


Figure 2-3 A Hierarchy of ADM Processes Example

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 <i>n</i>	Iteration 1	Iteration <i>n</i>	Iteration 1	Iteration <i>n</i>
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 <i>n</i>	Iteration 1	Iteration <i>n</i>	Iteration 1	Iteration <i>n</i>
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.	<p>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.</p> <p>Opportunities, solutions, and migration plans are also considered to drive out the focus for change and test feasibility.</p>

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.
	Iteration <i>n</i>	Agree the Transition Architecture, refining the identified improvement opportunities to fit.	This iteration also delivers a provisional Migration Plan. 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 <i>n</i>	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.

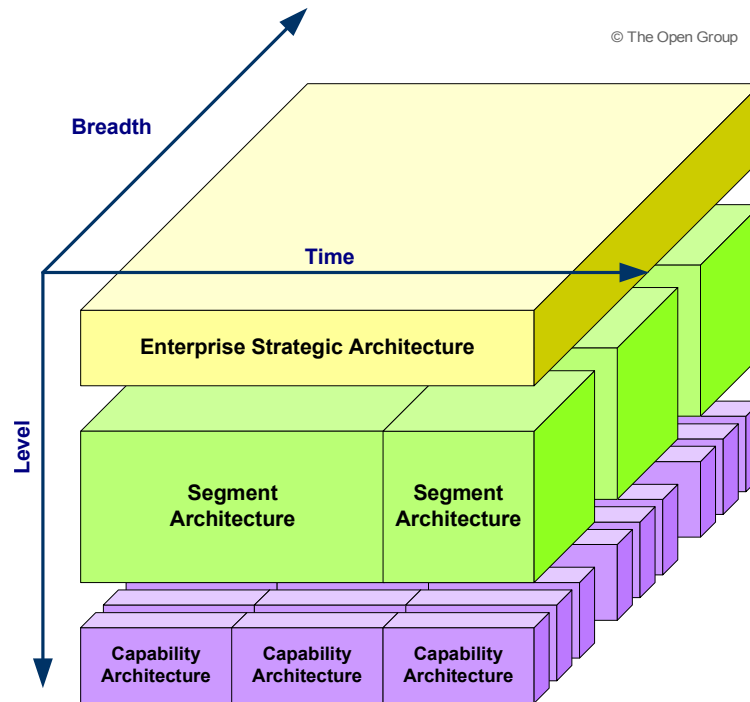


Figure 3-1 Summary Classification Model for Architecture Landscapes

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.

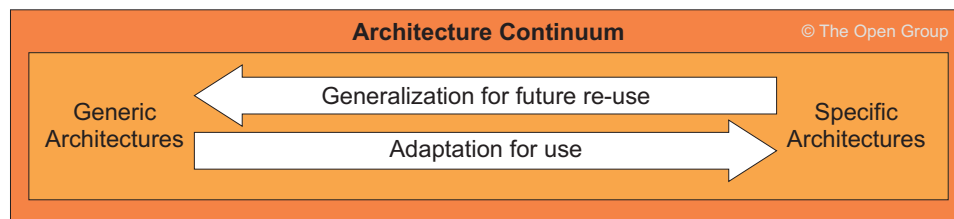


Figure 3-2 Summary of Architecture Continuum

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)	<p>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.</p> <p>Solution decomposition by subject matter is typically the fundamental technique for structuring both solutions and the architectures that represent them.</p>
Time	<p>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.</p>
Maturity/Volatility	<p>The maturity and volatility of a solution will typically impact the speed of execution required for the solution lifecycle.</p> <p>Additionally, volatility and maturity will shape investment priorities. Solutions existing in highly volatile environments may be better suited to rapid, agile development techniques.</p>

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

Characteristic	Usage to Support Architecture Partitioning
Depth	<p>The level of detail within an architecture has a strong correlation to the stakeholder groups that will be interested in the architecture.</p> <p>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.</p>

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.

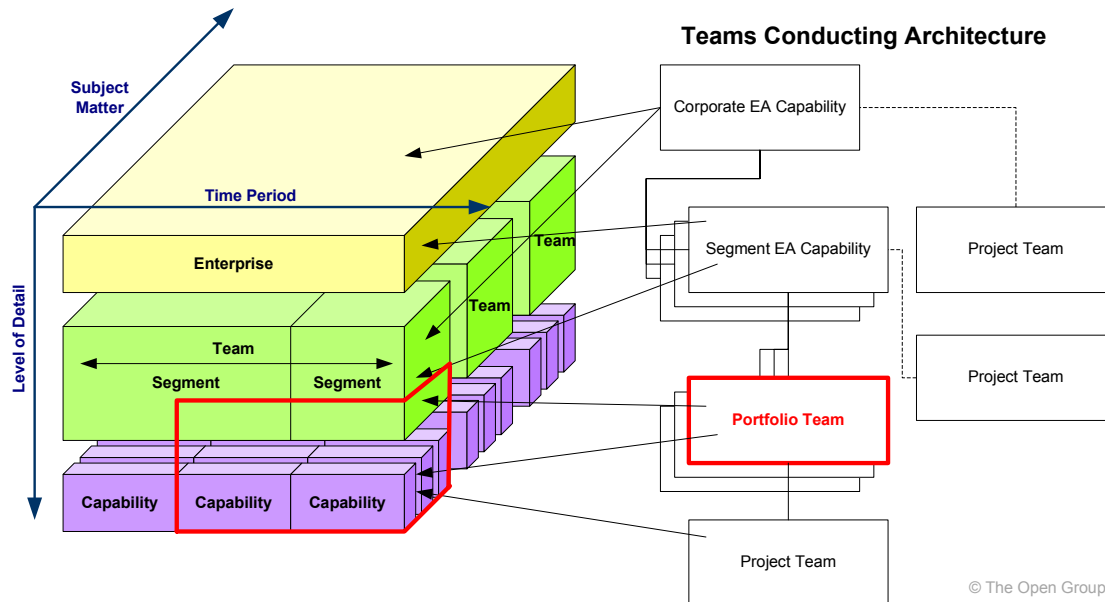


Figure 4-1 Allocation of Teams to Architecture Scope

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.

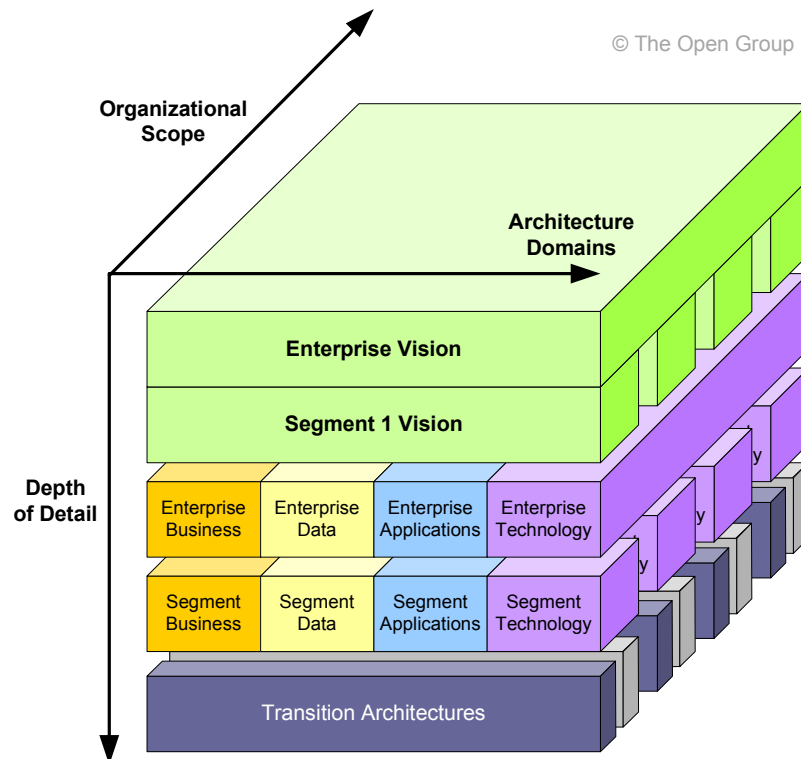


Figure 4-2 Architecture Content Aggregation

Index

ADM iteration	3
applying the ADM	17
Architecture Capability	5
architecture development	5, 9
architecture engagement	5
Architecture Governance	5
Architecture Landscape	17
architecture partitioning	21
architecture styles	1
baseline first	9
breadth	19
Capability Architecture	17
classification	21
depth	19
iteration	10
iteration cycles	4
iteration of the ADM	3
recency	20
Segment Architecture	17
Strategic Architecture	17
target first	9
time	19
Transition Planning	5

The Open Group Standard

TOGAF® Standard — Architecture Content

The Open Group

Copyright © 2005-2022, The Open Group

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

Any use of this publication for commercial purposes is subject to the terms of the Annual Commercial License relating to it. For further information, see www.opengroup.org/legal/licensing.

The Open Group Standard

TOGAF® Standard — Architecture Content

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

Chapter	1	Introduction	1
	1.1	Overview	1
	1.2	TOGAF Content Framework and Enterprise Metamodel	3
	1.2.1	Overview	3
	1.2.2	Content Framework	4
	1.2.3	Enterprise Metamodel	4
	1.2.4	The TOGAF Content Framework	5
	1.3	Content Framework and the TOGAF ADM	7
	1.4	The Enterprise Continuum	8
	1.5	The Architecture Repository	8
Chapter	2	TOGAF Content Framework and Enterprise Metamodel	9
	2.1	Overview	9
	2.2	TOGAF Enterprise Metamodel Vision	9
	2.2.1	Overview of the TOGAF Enterprise Metamodel	10
	2.3	TOGAF Enterprise Metamodel in Detail	11
	2.4	TOGAF Enterprise Metamodel Entities	12
	2.5	TOGAF Enterprise Metamodel Attributes	15
	2.6	TOGAF Enterprise Metamodel Relationships	25
Chapter	3	Architectural Artifacts	31
	3.1	Basic Concepts	31
	3.1.1	Simple Example of an Architecture Viewpoint and Architecture View	33
	3.2	Developing Architecture Views in the ADM	34
	3.2.1	General Guidelines	34
	3.2.2	Architecture View Creation Process	35
	3.3	Views, Tools, and Languages	36
	3.3.1	Overview	36
	3.4	Architecture Views and Architecture Viewpoints	36
	3.4.1	Example of Architecture Views and Architecture Viewpoints	36
	3.4.2	Architecture Views and Architecture Viewpoints in Enterprise Architecture	37
	3.4.3	Need for a Common Language and Interoperable Tools for Architecture Description	38
	3.5	Conclusions	38
	3.6	Architectural Artifacts by ADM Phase	39
	3.6.1	Preliminary Phase	41
	3.6.2	Phase A: Architecture Vision	42
	3.6.3	Phase B: Business Architecture	43
	3.6.4	Phase C: Data Architecture	49

3.6.5	Phase C: Application Architecture	52
3.6.6	Phase D: Technology Architecture	57
3.6.7	Phase E: Opportunities and Solutions	61
3.6.8	Requirements Management.....	61
Chapter 4	Architecture Deliverables	63
4.1	Introduction	63
4.2	Deliverable Descriptions	64
4.2.1	Architecture Building Blocks.....	65
4.2.2	Architecture Contract	65
4.2.3	Architecture Definition Document	66
4.2.4	Architecture Principles	67
4.2.5	Architecture Repository.....	68
4.2.6	Architecture Requirements Specification	68
4.2.7	Architecture Roadmap	69
4.2.8	Architecture Vision	70
4.2.9	Business Principles, Business Goals, and Business Drivers.....	70
4.2.10	Capability Assessment.....	71
4.2.11	Change Request	72
4.2.12	Communications Plan	73
4.2.13	Compliance Assessment	73
4.2.14	Implementation and Migration Plan.....	74
4.2.15	Implementation Governance Model	75
4.2.16	Organizational Model for Enterprise Architecture	75
4.2.17	Request for Architecture Work	76
4.2.18	Requirements Impact Assessment	76
4.2.19	Solution Building Blocks.....	77
4.2.20	Statement of Architecture Work	77
4.2.21	Tailored Architecture Framework.....	78
Chapter 5	Building Blocks.....	79
5.1	Overview	79
5.2	Introduction to Building Blocks	79
5.2.1	Overview	79
5.2.2	Generic Characteristics.....	79
5.2.3	Architecture Building Blocks.....	80
5.2.4	Solution Building Blocks.....	81
5.3	Building Blocks and the ADM.....	82
5.3.1	Basic Principles.....	82
5.3.2	Building Block Specification Process in the ADM	83
Chapter 6	Enterprise Continuum.....	85
6.1	Overview	85
6.2	Enterprise Continuum and Architecture Re-Use	85
6.3	Constituents of the Enterprise Continuum	86
6.4	Enterprise Continuum in Detail	87
6.4.1	Architecture Continuum.....	88
6.4.2	Solutions Continuum	91
6.5	The Enterprise Continuum and the ADM	93

Contents

6.6	The Enterprise Continuum and Your Organization	93
6.6.1	Relationships	93
6.6.2	Your Enterprise	95
Chapter 7	Architecture Repository	97
7.1	Overview	97
7.2	Architecture Landscape	98
7.3	Reference Library	99
7.3.1	Overview	99
7.4	Standards Library.....	100
7.4.1	Overview	100
7.4.2	Types of Standard	101
7.4.3	Standards Lifecycle.....	101
7.4.4	Standards Classification within the Standards Library	102
7.5	Governance Repository	103
7.5.1	Overview	103
7.5.2	Contents of the Governance Repository	103
7.6	The Architecture Requirements Repository	104
7.6.1	Overview	104
7.6.2	Contents of the Architecture Requirements Repository	104
7.7	Solutions Landscape.....	105
7.8	The Enterprise Repository	105
7.9	External Repositories.....	106
7.9.1	External Reference Models.....	106
7.9.2	External Standards	106
7.9.3	Architecture Board Approvals	106
	Index.....	107

List of Figures

1-1	Relationships between Deliverables, Artifacts, and Building Blocks	2
1-2	Example — Architecture Definition Document	2
1-3	Content Framework by ADM Phase	5
1-4	Content Framework Overview	6
2-1	Using the Content Framework to Structure the TOGAF Enterprise Metamodel	10
2-2	Relationships between Entities in the TOGAF Enterprise Metamodel.....	11
3-1	Basic Architectural Concepts.....	31
3-2	Example Architecture View — The Open Group Business Domains	33
3-3	Interactions between Metamodel, Building Blocks, Diagrams, and Stakeholders	39
3-4	Artifacts Associated with the Enterprise Metamodel	40
5-1	Key ADM Phases/Steps at which Building Blocks are Evolved/Specified	83

6-1	Enterprise Continuum.....	86
6-2	Architecture Continuum	88
6-3	Solutions Continuum	91
6-4	Relationships between Architecture and Solutions Continua	93
7-1	Overview of Architecture Repository	97
7-2	Architecture Continuum	99

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 — Architecture Content.

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.

Figure 3-1 is reprinted and adapted from Figure 2 of ISO/IEC/IEEE 42010:2011, Systems and Software Engineering — Architecture Description, with permission from IEEE®. Copyright© 2011, by IEEE. The IEEE disclaims any responsibility or liability resulting from the placement and use in the described manner.

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.

CMMI is a registered trademark of CMMI Institute.

Energistics is a registered trademark of Energistics in the United States.

IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers, Inc.

ITIL, MSP, and PRINCE2 are registered trademarks of AXELOS Limited.

Object Management Group, OMG, and UML are registered trademarks and BPMN, Business Process Modeling Notation, and Unified Modeling Language are trademarks of the Object Management Group.

PMBOK is a registered trademark of the Project Management Institute, Inc. which is registered in the United States and other nations.

Zachman is a registered trademark of Zachman International, Inc.

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 — Architecture Content (this document).

1.1 Overview

Architects executing the Architecture Development Method (ADM) will produce a number of outputs as a result of their efforts, such as process flows, architectural requirements, project plans, or project compliance assessments. The Content Framework provides a structural model for architectural content that allows the major work products that an architect creates to be consistently defined, structured, and presented.

The Content Framework provided here is intended to allow the TOGAF framework to be used as a stand-alone framework for architecture within an enterprise. However, other Content Frameworks exist (such as the Zachman[®] Framework) and it is anticipated that some enterprises may opt to use an external framework in conjunction with the TOGAF framework. In these cases, the TOGAF Content Framework provides a useful reference and starting point for TOGAF content to be mapped to other Content Frameworks.

The Architecture Content Framework uses the following three categories to describe the type of architectural work product within the context of use:

- A **deliverable** is a work product that is contractually specified and in turn formally reviewed, approved, and signed off by the stakeholders

Deliverables represent the output of projects and those deliverables that are in documentation form will typically be archived at completion of a project, or transitioned into an Architecture Repository as a reference model, standard, or snapshot of the Architecture Landscape at a point in time.

- An **artifact** is an architectural work product that describes an aspect of the architecture

Artifacts are generally classified as catalogs (lists of things), matrices (showing relationships between things), and diagrams (pictures of things). Examples include a requirements catalog, application interaction matrix, and a value chain diagram. An architectural deliverable may contain many artifacts and artifacts will form the content of the Architecture Repository.

- A **building block** represents a potentially re-usable component of enterprise capability that can be combined with other building blocks to deliver architectures and solutions

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

- **Architecture Building Blocks (ABBs)** typically describe what is required of SBBs at a more logical or supplier-independent level; those requirements may include services to be performed, data resources, and capabilities needed. ABBs include logical business, application, and technology components
- **Solution Building Blocks (SBBs)** represent physical or supplier-specific components that have the capability to realize part or all of a more logical ABB. There are business, application, and technology SBBs.

The relationships between deliverables, artifacts, and building blocks are shown in [Figure 1-1](#).

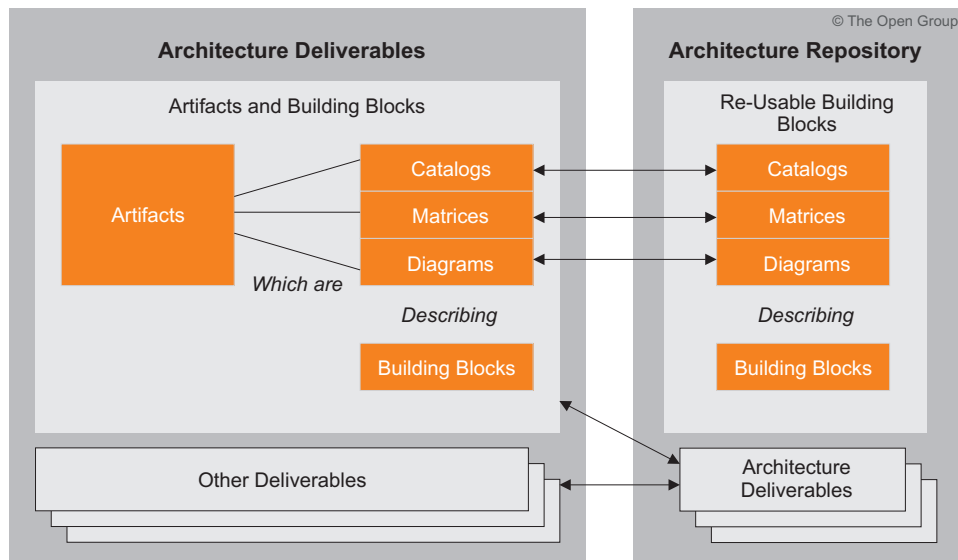


Figure 1-1 Relationships between Deliverables, Artifacts, and Building Blocks

For example, an Architecture Definition Document is a deliverable that documents an Architecture Description. This document will contain a number of complementary artifacts that are architecture views of the building blocks relevant to the architecture. For example, a process flow diagram (an artifact) may be created to describe the target call handling process (a building block). This artifact may also describe other building blocks, such as the actors involved in the process (e.g., a Customer Services Representative). An example of the relationships between deliverables, artifacts, and building blocks is illustrated in [Figure 1-2](#).

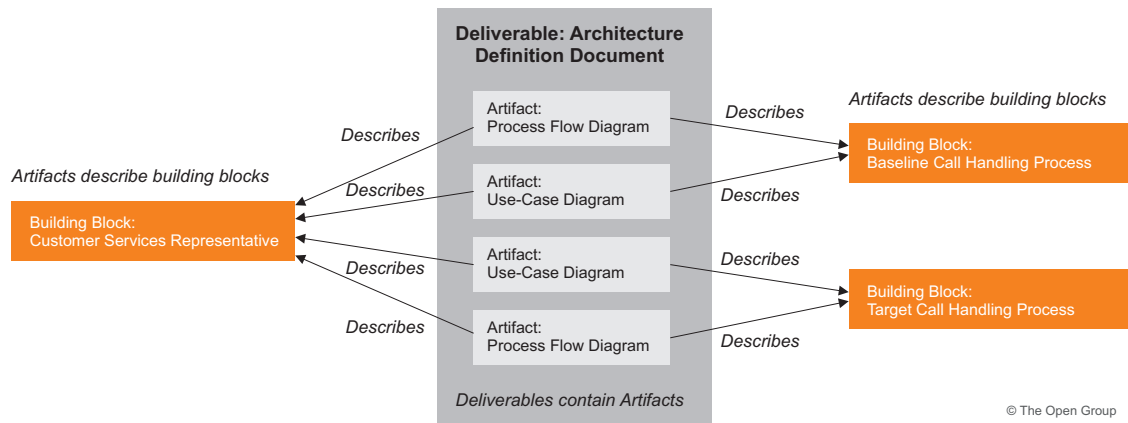


Figure 1-2 Example — Architecture Definition Document

1.2 TOGAF Content Framework and Enterprise Metamodel

1.2.1 Overview

The TOGAF ADM provides lifecycle management to create and manage architectures within an enterprise. At each phase within the ADM, a discussion of inputs, outputs, and steps describes a number of architecture work products.

An essential task when establishing the enterprise-specific Enterprise Architecture Capability in the Preliminary Phase of the ADM is to define:

- A categorization framework to be used to structure the Architecture Descriptions, the work products used to express an architecture, and the collection of models that describe the architecture; this is referred to as the **Content Framework**
- An understanding of the types of entities within the enterprise and the relationships between them that need to be captured, stored, and analyzed in order to create the Architecture Description; this **Enterprise Metamodel** depicts this information in the form of a formal model
- The specific artifacts to be developed (see [Chapter 4](#))

The Content Framework chosen is likely to be influenced by:

- The Architecture Framework selected as the basis for the Enterprise Architecture Capability
- The chosen software tool used to support the Enterprise Architecture Capability

1.2.2 Content Framework

The Content Framework defines a categorization framework to be used to structure the Architecture Description, the work product used to express an architecture, and the collection of models that describe the architecture.

The Architecture Repository, which is explained in [Section 4.2.5](#), is structured to store the artifacts and work products identified in the Content Framework. The Content Framework is one element of the Enterprise-Specific Architecture Framework.

1.2.3 Enterprise Metamodel

The TOGAF Standard encourages development of an Enterprise Metamodel, which defines the types of entity to appear in the models that describe the enterprise, together with the relationships between these entities.

An Enterprise Metamodel provides value in several ways:

- It gives architects a starter set of the types of thing to investigate and to cover in their models
- It provides a form of completeness-check for any architecture modeling language, or architecture metamodel, that is proposed for use in an enterprise

Namely, how completely does it handle the types of entity in the Enterprise Metamodel, and manage required facts about them such as their attributes and relationships?

- It can help ensure:
 - Consistency
 - Completeness
 - Traceability

Note that the TOGAF Standard does not aim to constrain an enterprise's:

- Selection of artifacts
- Modeling notation

The TOGAF Standard may use the ArchiMate[®] modeling language, Business Process Modeling Notation[™] (BPMN[™]), Unified Modeling Language[™] (UML[®]), entity relationship diagramming, flowcharting, or any other notation that can express some TOGAF ideas.

The types of entity within an enterprise and the relationships between them are specific to the individual enterprise. Developing a high-quality metamodel is an important aspect of establishing the Enterprise Architecture Capability.

1.2.4 The TOGAF Content Framework

The TOGAF Content Framework defines a categorization framework to be used to structure the Architecture Description, the work products used to express an architecture, and the collection of models that describe the architecture.

There are many alternative Content Frameworks (e.g., the TOGAF Content Framework, the Zachman Framework, DoDAF, NAF, etc.). Selecting a Content Framework is essential even though the choice of Content Framework is less important. The final Content Framework is usually adapted to fit specific organization needs.

The TOGAF Content Framework is intended to:

- Provide a detailed model of architectural work products
- Drive consistency in the outputs created when following the ADM
- Provide a comprehensive checklist of architecture output that could be created
- Reduce the risk of gaps within the final architecture deliverable set
- Help an enterprise mandate standard architecture concepts, terms, and deliverables

At the highest level, the TOGAF Content Framework (see [Figure 1-3](#)) is structured in line with the phases of the ADM.

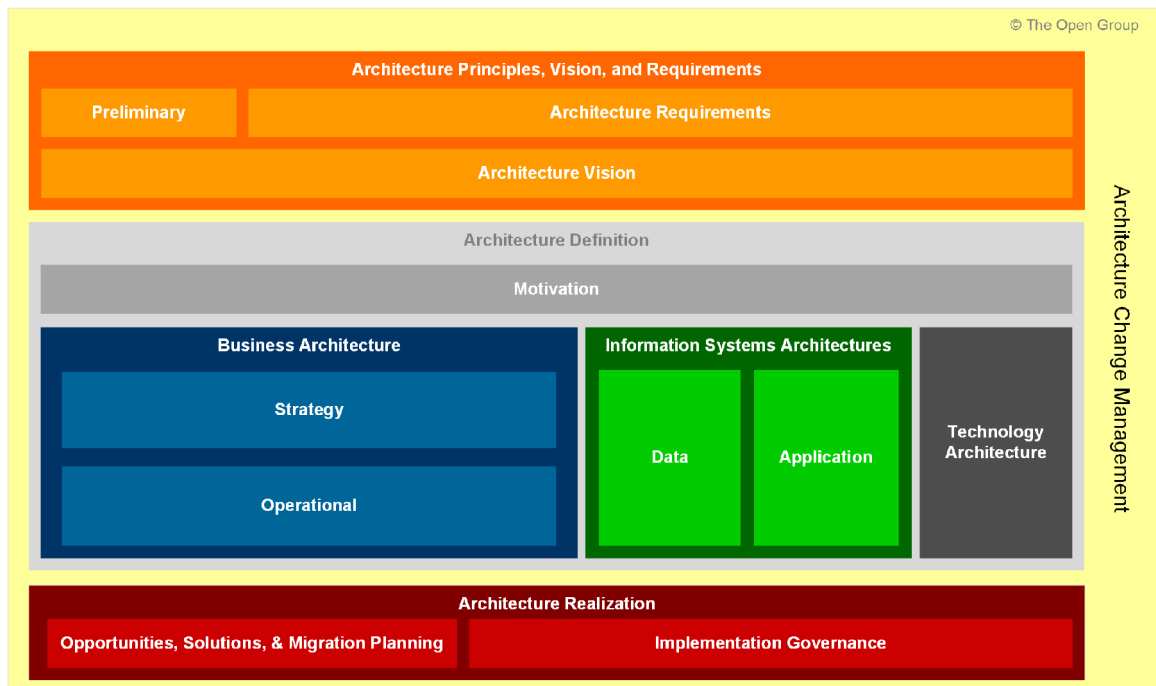


Figure 1-3 Content Framework by ADM Phase

- **Architecture Principles, Vision, Motivation, and Requirements** models are intended to capture the surrounding context of formal architecture models, including general Architecture Principles, strategic context that forms input for architecture modeling, and requirements generated from the architecture

The relevant aspects of the business context that have given rise to the Request for Architecture work are typically investigated, refined, validated, and recorded in the Preliminary and Architecture Vision phases.

- **Business Architecture** captures architecture models of the business, looking specifically at factors that motivate the enterprise, its structure, and its capabilities
- **Information Systems Architecture** models capture architecture models of IT systems, looking at applications and data in line with the TOGAF ADM phases
- **Technology Architecture** models capture technology assets that are used to implement and realize information system solutions
- **Architecture Realization/Transformation** models capture change roadmaps showing transition between architecture states and binding statements that are used to steer and govern an implementation of the architecture
- **Architecture Change Management** models capture value realization management events, internal and external, that impact the Enterprise Architecture and the generation of requirements for action

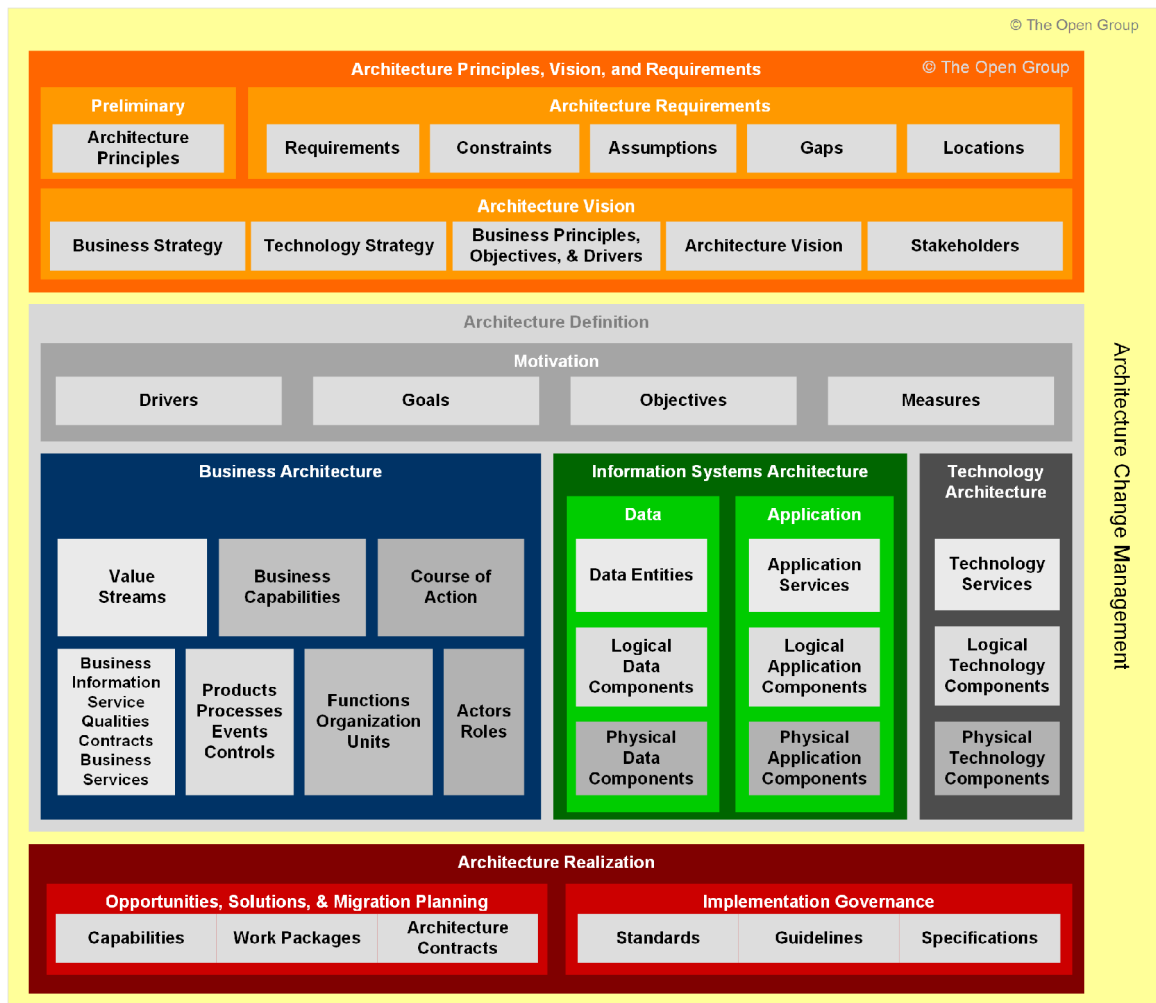


Figure 1-4 Content Framework Overview

1.3 Content Framework and the TOGAF ADM

The TOGAF ADM describes the process of moving from a baseline state of the enterprise to a target state of the enterprise. The ADM will address a business need through a process of visioning, architecture definition, transformation planning, and Architecture Governance. At each stage in this process, the ADM requires information as inputs and will create outputs as a result of executing a number of steps. The Content Framework provides an underlying structure for the ADM that defines inputs and outputs in more detail and puts each deliverable into the context of the holistic architecture view of the enterprise.

The Content Framework should therefore be used as a companion to the ADM. The ADM describes what needs to be done to create an architecture and the Content Framework describes what the architecture should look like once it is done.

1.4 The Enterprise Continuum

It is usually impossible to create a single unified architecture that meets all the requirements of all stakeholders for all time. Therefore, the Enterprise Architect will need to deal not just with a single Enterprise Architecture, but with many related Enterprise Architectures.

Each architecture may have a different purpose and architectures may relate to one another. Effectively bounding the scope of an architecture is therefore a Critical Success Factor (CSF) in allowing architects to break down a complex problem space into manageable components that can be individually addressed.

The Enterprise Continuum provides a view of the Architecture Repository that shows the evolution of these related architectures from generic to specific, from abstract to concrete, and from logical to physical.

[Chapter 6](#) discusses the Enterprise Continuum; including the Architecture Continuum and the Solutions Continuum.

1.5 The Architecture Repository

Operating a mature Architecture Capability within a large enterprise creates a huge volume of architectural output. Effective management and leverage of these architectural work products require a formal taxonomy for different types of architectural asset alongside dedicated processes and tools for architectural content storage.

[Chapter 7](#) provides a structural framework for an Architecture Repository that allows an enterprise to distinguish between different types of architectural assets that exist at different levels of abstraction in the organization.

Chapter 2: TOGAF Content Framework and Enterprise Metamodel

2.1 Overview

The TOGAF ADM provides a process lifecycle to create and manage architectures within an enterprise. At each phase within the ADM, a discussion of inputs, outputs, and steps describes a number of architectural work products or artifacts, such as process and application.

The Content Framework and Enterprise Metamodel provided here define a formal structure for these terms to ensure consistency within the ADM and also to provide guidance for organizations that wish to implement their architecture within an architecture tool.

The **Content Framework** defines a categorization framework to be used to structure the Architecture Description, the work product used to express an architecture, and the collection of models that describe the architecture.

The **Enterprise Metamodel** defines the types of entities to appear in the models that describe the enterprise, together with the relationships between these entities.

2.2 TOGAF Enterprise Metamodel Vision

The TOGAF Standard includes the TOGAF Enterprise Metamodel which captures the entities and relationships that are likely to be encountered in the majority of enterprises. This may be used as the basis for developing an Organization-Specific Metamodel when establishing the Enterprise Architecture Capability in the Preliminary Phase and also provides the context for the specific artifacts referenced in the descriptions of the ADM phases and described in detail in [Chapter 3](#).

When developing an Organization-Specific Metamodel, architects may choose not to include entities and relationships from the TOGAF Enterprise Metamodel which are not relevant and/or add additional entities and relationships.

This section provides an overview of the TOGAF Enterprise Metamodel. Subsequent sections discuss each area of the metamodel in more detail.

2.2.1 Overview of the TOGAF Enterprise Metamodel

The TOGAF Enterprise Metamodel includes a set of entities, defined in [Section 2.4](#), that allow architectural concepts to be captured, stored, filtered, queried, and represented in a way that supports consistency, completeness, and traceability.

The categorization mechanism of the Content Framework may be used to structure a representation of the TOGAF Enterprise Metamodel, as shown in [Figure 2-1](#).

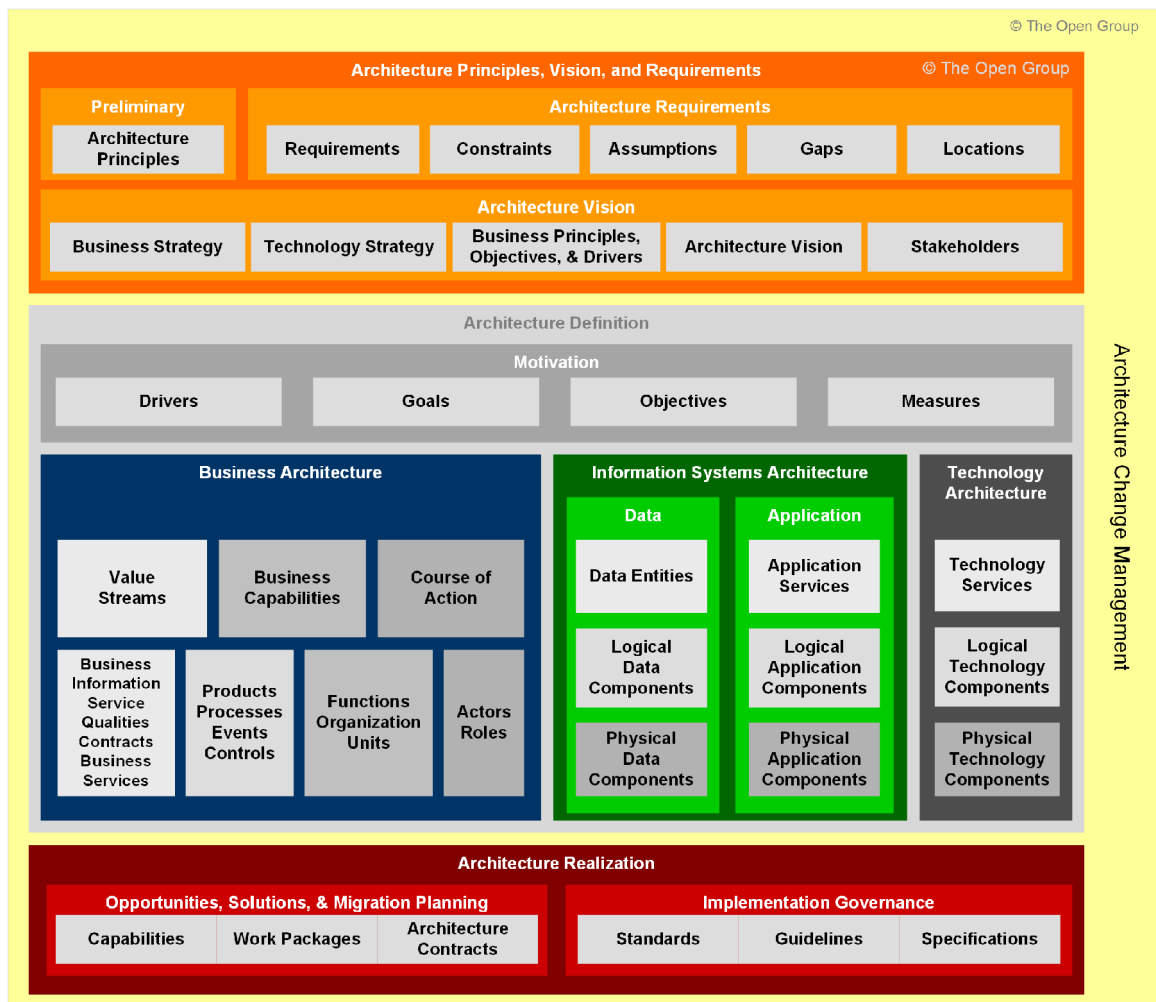


Figure 2-1 Using the Content Framework to Structure the TOGAF Enterprise Metamodel

2.3 TOGAF Enterprise Metamodel in Detail

The relationships between entities in the TOGAF Enterprise Metamodel are shown in [Figure 2-2](#).

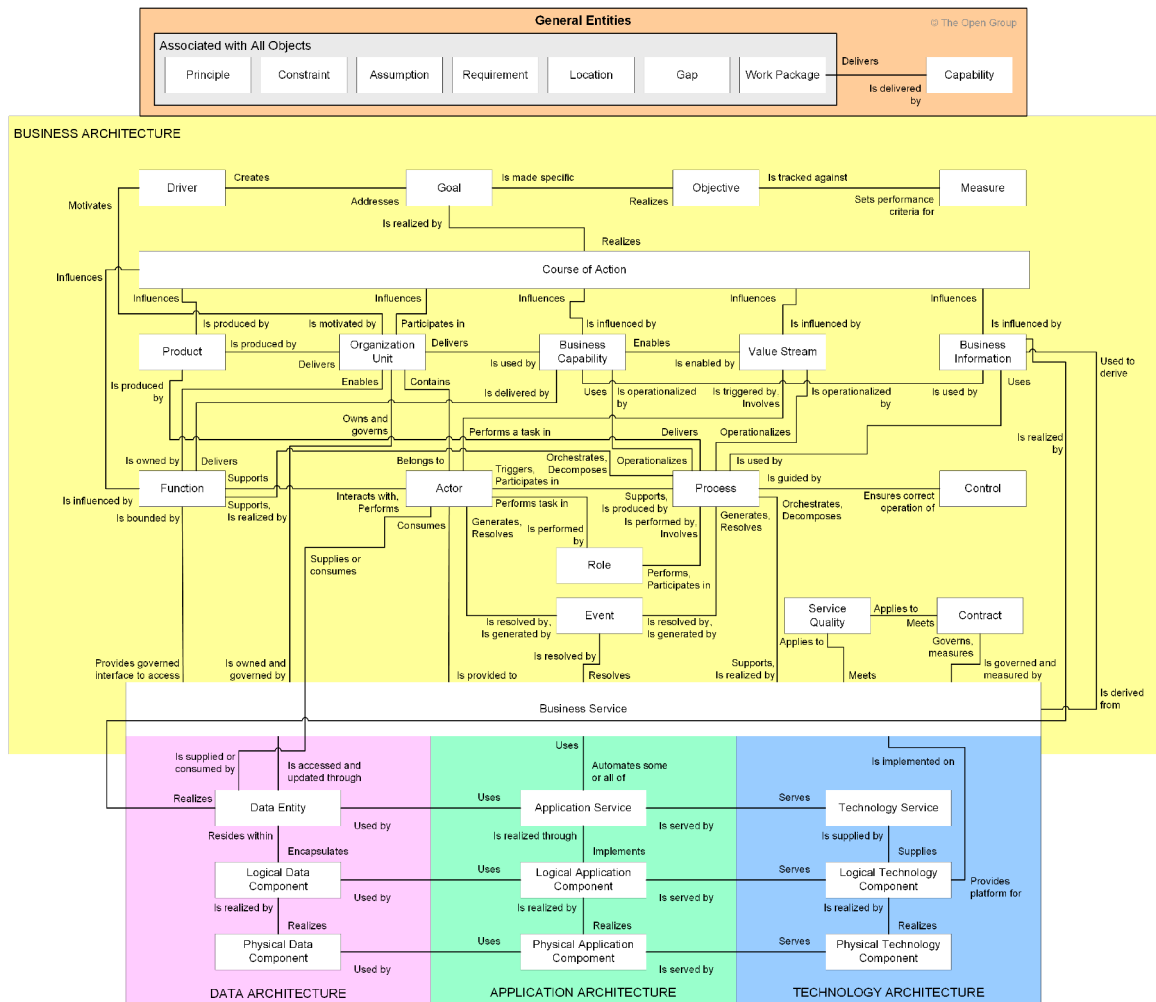


Figure 2-2 Relationships between Entities in the TOGAF Enterprise Metamodel

2.4 TOGAF Enterprise Metamodel Entities

The following table lists and describes the entities within the Enterprise Metamodel.

Metamodel Entity	Description
Actor	A person, organization, or system that has a role that initiates or interacts with activities; for example, a sales representative who travels to visit customers. Actors may be internal or external to an organization. In the automotive industry, an original equipment manufacturer would be considered an actor by an automotive dealership that interacts with its supply chain activities.
Application Service	The automated elements of a business service. An application service may deliver or support part or all of one or more business services.
Assumption	A statement of probable fact that has not been fully validated at this stage, due to external constraints. For example, it may be assumed that an existing application will support a certain set of functional requirements, although those requirements may not yet have been individually validated.
Business Capability	A particular ability that a business may possess or exchange to achieve a particular purpose.
Business Information	Represents a concept and its semantics used within the business.
Business Service	Supports the business by encapsulating a unique element of business behavior; a service offered external to the enterprise may be supported by business services.
Capability	An ability that an organization, person, or system possesses. Note: This is a general-purpose definition. See <i>Business Capability</i> for how this concept is refined for usage in Business Architecture.
Constraint	An external factor that prevents an organization from pursuing particular approaches to meet its goals. For example, customer data is not harmonized within the organization, regionally or nationally, constraining the organization's ability to offer effective customer service.
Contract	An agreement between a consumer and a provider that establishes functional and non-functional parameters for interaction. This applies to all types of service interactions within the metamodel.
Control	A decision-making step with accompanying decision logic used to determine execution approach for a process or to ensure that a process complies with governance criteria. For example, a sign-off control on the purchase request processing process that checks whether the total value of the request is within the sign-off limits of the requester, or whether it needs escalating to higher authority.
Course of Action	Direction and focus provided by strategic goals and objectives, often to deliver the value proposition characterized in the business model.
Data Entity	Represents data that is recognized by the business as a distinct concept.

Metamodel Entity	Description
Driver	An external or internal condition that motivates the organization to define its goals. An example of an external driver is a change in regulation or compliance rules which, for example, require changes to the way an organization operates; i.e., Sarbanes-Oxley in the US.
Event	An organizational state change that triggers processing events; may originate from inside or outside the organization and may be resolved inside or outside the organization.
Function	A set of business behaviors based on a chosen set of criteria. Functions are usually close-coupled to/with organizational units.
Gap	A statement of difference between two states. Used in the context of gap analysis, where the difference between the Baseline and Target Architecture is identified. Note: Gap analysis is described in the TOGAF Standard — ADM Techniques.
Goal	A high-level statement of intent or direction for an organization. Typically used to measure success of an organization.
Location	A place where activities occur. Locations can be composed and decomposed.
Logical Application Component	An encapsulation of application functionality that is definable by services offered and data maintained, independently of implementation and technology.
Logical Data Component	A data structure composed of logically-related data entities.
Logical Technology Component	An implementation-independent encapsulation of technology services.
Measure	An indicator or factor that can be tracked, usually on an ongoing basis, to determine success or alignment with objectives and goals.
Objective	An organizational aim that is declared in a Simple, Measurable, Actionable, Realistic, and Timebound (SMART) way. For example, "Increase capacity utilization by 30% by the end of the year, to support the planned increase in market share".
Organization Unit	A self-contained unit of resources with goals, objectives, and measures. Organization units may include external parties and business partner organizations.
Physical Application Component	A realization of logical application functionality using components of functionality in applications that may be hired, procured, or built.
Physical Data Component	A data structure that realizes related logical data components represented in the format or schema required by a particular technology.
Physical Technology Component	A realization of logical technology functionality using a particular technology product that may be deployed.

Metamodel Entity	Description
Principle	<p>A qualitative statement of intent that should be met by the architecture. It has at least a supporting rationale and a measure of importance.</p> <p>Note: A sample set of Architecture Principles is defined in the TOGAF Standard — ADM Techniques.</p>
Process	<p>A process represents a sequence of activities that together achieve a specified outcome, can be decomposed into sub-processes, and can show operation of a business capability or service (at next level of detail).</p> <p>Processes may also be used to link organizations, business capabilities, services, and processes. A process may realize one service and/or orchestrate subordinate services.</p>
Product	An outcome generated by the business to be offered to customers. Products include materials and/or services.
Requirement	A quantitative statement of business need that must be met by a particular architecture or work package.
Role	<p>The usual or expected behavior of an actor, or the part somebody or something plays in a particular process or event. An actor may have a number of roles.</p> <p>See also <i>Actor</i>.</p>
Service Quality	A configuration of non-functional requirements or attributes that may be assigned to a business, application, or technology service.
Technology Service	A technical capability required to provide enabling infrastructure that supports the delivery of applications.
Value Stream	A representation of an end-to-end collection of activities that create an overall result for a customer, stakeholder, or end-user.
Work Package	A set of actions identified to achieve one or more objectives for the business. A work package can be a part of a project, a complete project, or a program.

2.5 TOGAF Enterprise Metamodel Attributes

The following table shows typical attributes for each of the metamodel entities described previously.

Metamodel Entity	Attribute	Description
All Metamodel Entities	ID	Unique identifier for the architecture entity.
	Name	Brief name of the architecture entity.
	Description	Textual description of the architecture entity.
	Category	User-definable categorization taxonomy for each metamodel entity.
	Source	Location from where the information was collected.
	Owner	Owner of the architecture entity.
Capability	Business value	Describes how this capability provides value to the enterprise.
	Increments	Lists possible maturity/quality levels for the capability.
Constraint	No additional attributes	This metamodel entity has only basic attributes.
Gap	No additional attributes	This metamodel entity has only basic attributes.
Location	Category	The following categories of Location apply: Region (applies to a grouping of countries or territory; e.g., South East Asia, UK, and Ireland), Country (applies to a single country; e.g., US), Building (applies to a site of operation; where several offices are collected in a single city, this category may represent a city), and Specific Location (applies to any specific location within a building, such as a server room). The nature of the business may introduce other Locations: Ship or Port for a ferry company, Mine for a gold company, Car for a police force, Hotel for any firm's traveling workers, and so on.

Metamodel Entity	Attribute	Description
Principle	Category	The following categories of principle apply: Guiding Principle, Business Principle, Data Principle, Application Principle, Integration Principle, Technology Principle.
	Priority	Priority of this principle relative to other principles.
	Statement of principle	Statement of what the principle is.
	Rationale	Statement of why the principle is required and the desired outcome to be reached.
	Implication	Statement of what the principle means in practical terms.
	Metric	Identifies mechanisms that will be used to measure whether the principle has been met or not.
Requirement	Statement of requirement	Statement of what the requirement is, including a definition of whether the requirement shall be met, should be met, or may be met.
	Rationale	Statement of why the requirement exists.
	Acceptance criteria	The parameters that will be fulfilled if the requirement is being met, together with the tests that will be carried out to assess the state of the parameters.
Actor	# FTEs	Estimated number of FTEs that operate as this actor.
	Actor goal	Objectives that this actor has, in general terms.
	Actor tasks	Tasks that this actor performs, in general terms.
Business Service	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
	Standard creation date	If the business service is a standard, when the standard was created.
	Last standard review date	Last date that the standard was reviewed.

Metamodel Entity	Attribute	Description
	Next standard review date	Next date for the standard to be reviewed.
	Retire date	Date when the standard was/will be retired.
Contract	Behavior characteristics	Functional behavior to be supported within the scope of the contract.
	Service name "caller"	Consuming service.
	Service name "called"	Providing service.
	Service quality characteristics	Non-functional behavior to be supported within the scope of the contract.
	Availability characteristics	Degree to which something is available for use.
	Service times	Hours during which the service must be available.
	Manageability characteristics	Ability to gather information about the state of something and control it.
	Serviceability characteristics	Ability to identify problems and take corrective action, such as to repair or upgrade a component in a running system.
	Performance characteristics	Ability of a component to perform its tasks in an appropriate time.
	Response requirements	Response times that the service provider must meet for particular operations.
	Reliability characteristics	Resistance to failure.
	Quality of information required	Contracted requirements on accuracy and completeness of information.
	Contract control requirements	Level of governance and enforcement applied to the contractual parameters for overall service.
	Result control requirements	Measures in place to ensure that each service request meets contracted criteria.
	Recoverability characteristics	Ability to restore a system to a working state after an interruption.
	Locatability characteristics	Ability of a system to be found when needed.

Metamodel Entity	Attribute	Description
	Security characteristics	Ability of a system to prevent unauthorized access to functions and data.
	Privacy characteristics	Protection of data from unauthorized access.
	Integrity characteristics	Ability of a system to ensure that data has not been corrupted.
	Credibility characteristics	Ability of a system to ensure that the service request originates from an authorized source.
	Localization characteristics	Ability of a service to support localized variants for different consumer groups.
	Internationalization characteristics	Ability of a service to support international variations in business logic and data representation (such as character set).
	Interoperability characteristics	Ability of the service to interoperate with different technical environments, inside and outside of the organization.
	Scalability characteristics	Ability of the service to grow or shrink its performance or capacity appropriately to the demands of the environment in which it operates.
	Portability characteristics	Of data, people, applications, and components.
	Extensibility characteristics	Ability to accept new functionality.
	Capacity characteristics	Contracted capacity of the service provider to meet requests.
	Throughput	Required throughput capacity.
	Throughput period	Time period needed to deliver throughput capacity.
	Growth	Expected future growth rate of service request.
	Growth period	Time period needed to reach the expected growth rate.
	Peak profile short term	Short-term profile of peak service traffic.
	Peak profile long term	Long-term profile of peak service traffic.

Metamodel Entity	Attribute	Description
Control	No additional attributes	This metamodel entity has only basic attributes.
Driver	No additional attributes	This metamodel entity has only basic attributes.
Event	No additional attributes	This metamodel entity has only basic attributes.
Function	Standards class Standard creation date Last standard review date Next standard review date Retire date	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard. If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed. Date when the standard was/will be retired.
Goal	No additional attributes	This metamodel entity has only basic attributes.
Measure	No additional attributes	This metamodel entity has only basic attributes.
Objective	No additional attributes	This metamodel entity has only basic attributes.
Organization Unit	Headcount	Number of FTEs working within the organization.
Process	Standards class Standard creation date Last standard review date Next standard review date Retire date Process criticality Manual or automated Process volumetrics	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard. If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed. Date when the standard was/will be retired. Criticality of this process to business operations. Whether this process is supported by IT or is a manual process. Data on frequency of process execution.

Metamodel Entity	Attribute	Description
Product	No additional attributes	This metamodel entity has only basic attributes.
Role	Estimated number of FTEs that operate in this Role	This metamodel entity has only basic attributes.
Service Quality	No additional attributes	This metamodel entity has only basic attributes.
Service	Standards class Standard creation date Last standard review date Next standard review date Retire date	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard. If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed. Date when the standard was/will be retired.
Application Component	Standards class Standard creation date Last standard review date Next standard review date Retire date	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard. If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed. Date when the standard was/will be retired.
Application Service	Standards class Standard creation date Last standard review date Next standard review date Retire date	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard. If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed. Date when the standard was/will be retired.
Logical Application Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.

Metamodel Entity	Attribute	Description
	Standard creation date Last standard review date Next standard review date Retire date	If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed. Date when the standard was/will be retired.
Physical Application Component	Lifecycle status Standards class Standard creation date Last standard review date Next standard review date Retire date Initial live date Date of last release Date of next release Retirement date Availability characteristics Service times Manageability characteristics Serviceability characteristics Performance characteristics	Proposed, In Development, Live, Phasing Out, Retired. Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard. If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed. Date when the standard was/will be retired. Date when the first release of the application was/will be released into production. Date when the last release of the application was released into production. Date when the next release of the application will be released into production. Date when the application was/will be retired. Degree to which something is available for use. Hours during which the application must be available. Ability to gather information about the state of something and control it. Ability to identify problems and take corrective action, such as to repair or upgrade a component in a running system. Ability of a component to perform its tasks in an appropriate time.

Metamodel Entity	Attribute	Description
	Reliability characteristics	Resistance to failure.
	Recoverability characteristics	Ability to restore a system to a working state after an interruption.
	Locatability characteristics	Ability of a system to be found when needed.
	Security characteristics	Ability of a system to prevent unauthorized access to functions and data.
	Privacy characteristics	Protection of data from unauthorized access.
	Integrity characteristics	Ability of a system to ensure that data has not been corrupted.
	Credibility characteristics	Ability of a system to ensure that the service request originates from an authorized source.
	Localization characteristics	Ability of a service to support localized variants for different consumer groups.
	Internationalization characteristics	Ability of a service to support international variations in business logic and data representation (such as character set).
	Interoperability characteristics	Ability of the service to interoperate with different technical environments, inside and outside of the organization.
	Scalability characteristics	Ability of the service to grow or shrink its performance or capacity appropriately to the demands of the environment in which it operates.
	Portability characteristics	Of data, people, applications, and components.
	Extensibility characteristics	Ability to accept new functionality.
	Capacity characteristics	Contracted capacity of the service provider to meet requests.
	Throughput	Required throughput capacity.
	Throughput period	Time period needed to deliver throughput capacity.
	Growth	Expected future growth rate of service request.

Metamodel Entity	Attribute	Description
	Growth period Peak profile short term Peak profile long term	Time period needed to reach the expected growth rate. Short-term profile of peak service traffic. Long-term profile of peak service traffic.
Data Entity	Category Privacy classification Retention classification	The following categories of data entity apply: Message, Internally Stored Entity. Level of restriction placed on access to the data. Level of retention to be placed on the data.
Logical Data Component	Standards class Standard creation date Last standard review date Next standard review date Retire date	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard. If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed. Date when the standard was/will be retired.
Physical Data Component	Standards class Standard creation date Last standard review date Next standard review date Retire date	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard. If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed. Date when the standard was/will be retired.
Logical Technology Component	Standards class Standard creation date Last standard review date Next standard review date	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard. If the product is a standard, when the standard was created. Last date that the standard was reviewed. Next date for the standard to be reviewed.

Metamodel Entity	Attribute	Description
	Retire date	Date when the standard was/will be retired.
	Category	Logical Technology Components are categorized according to the defined taxonomy (such as the TOGAF Technical Reference Model (TRM)), adapted to meet the needs of an individual organization.
Physical Technology Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
	Standard creation date	If the product is a standard, when the standard was created.
	Last standard review date	Last date that the standard was reviewed.
	Next standard review date	Next date for the standard to be reviewed.
	Retire date	Date when the standard was/will be retired.
	Category	Physical Technology Components are categorized according to the defined taxonomy (such as the TOGAF TRM), adapted to meet the needs of an individual organization.
	Product name	Name of the product making up the technology component.
	Module name	Module, or other sub-product, name making up the technology component.
	Vendor	Vendor providing the technology component.
	Version	Version of the product making up the technology component.
Technology Service	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.

Metamodel Entity	Attribute	Description
	Category	Technology Services are categorized according to the defined taxonomy (such as the TOGAF TRM), adapted to meet the needs of an individual organization.
Business Capability	No additional attributes	This metamodel entity has only basic attributes.
Technology Component	Standards class	Non-Standard, Proposed Standard, Provisional Standard, Standard, Phasing-Out Standard, Retired Standard.
Course of Action	No additional attributes	This metamodel entity has only basic attributes.
Value Stream	No additional attributes	This metamodel entity has only basic attributes.
Work Package	Category Capability delivered	The following categories of work package apply: Work Package, Work Stream, Project, Program, Portfolio. Describes the contribution this work package makes to capability delivery.

2.6 TOGAF Enterprise Metamodel Relationships

Source Entity	Target Entity	Name
Actor	Actor	Decomposes
Actor	Business Service	Consumes
Actor	Data Entity	Supplies or consumes
Actor	Event	Generates
Actor	Event	Resolves
Actor	Function	Interacts with
Actor	Function	Performs
Actor	Organization Unit	Belongs to
Actor	Process	Participates in
Actor	Process	Triggers
Actor	Role	Performs task in
Actor	Value Stream	Performs a task in
Application Service	Business Service	Automates some or all of
Application Service	Data Entity	Used by
Application Service	Logical Application Component	Is realized through

Source Entity	Target Entity	Name
Application Service	Technology Service	Is served by
Business Capability	Business Information	Uses
Business Capability	Course of Action	Is influenced by
Business Capability	Function	Is delivered by
Business Capability	Organization Unit	Is used by
Business Capability	Process	Is operationalized by
Business Capability	Value Stream	Enables
Business Information	Business Capability	Is used by
Business Information	Business Service	Used to derive
Business Information	Course of Action	Is influenced by
Business Information	Data Entity	Is realized by
Business Information	Process	Uses
Business Service	Actor	Is provided to
Business Service	Application Service	Uses
Business Service	Business Information	Is derived from
Business Service	Business Service Quality	Meets
Business Service	Business Service	Consumes
Business Service	Business Service	Decomposes
Business Service	Contract	Is governed and measured by
Business Service	Data Entity	Is accessed and updated through
Business Service	Event	Resolves
Business Service	Function	Provides governed interface to access
Business Service	Logical Technology Component	Is implemented on
Business Service	Organization Unit	Is owned and governed by
Business Service	Process	Is realized by
Business Service	Process	Supports
Capability	Work Package	Is delivered by
Contract	Business Service	Governs, measures
Contract	Service Quality	Meets
Control	Process	Ensures correct operation of
Course of Action	Business Capability	Influences
Course of Action	Business Information	Influences
Course of Action	Function	Influences
Course of Action	Goal	Realizes
Course of Action	Organization Unit	Influences
Course of Action	Product	Influences
Course of Action	Value Stream	Influences
Data Entity	Actor	Is supplied or consumed by
Data Entity	Application Service	Used by
Data Entity	Business Information	Realizes
Data Entity	Business Service	Is accessed and updated through
Data Entity	Data Entity	Decomposes
Data Entity	Data Entity	Relates to
Data Entity	Logical Data Component	Resides within

Source Entity	Target Entity	Name
Driver	Driver	Decomposes
Driver	Goal	Creates
Driver	Organization Unit	Motivates
Event	Actor	Is generated by
Event	Actor	Is resolved by
Event	Business Service	Is resolved by
Event	Process	Is generated by
Event	Process	Is resolved by
Function	Actor	Supports
Function	Business Capability	Delivers
Function	Business Service	Is bounded by
Function	Course of Action	Is influenced by
Function	Function	Communicates with
Function	Function	Decomposes
Function	Organization Unit	Is owned by
Function	Process	Orchestrates
Function	Process	Decomposes
Goal	Course of Action	Is realized by
Goal	Driver	Addresses
Goal	Goal	Decomposes
Goal	Objective	Is made specific
Logical Application Component	Application Service	Implements
Logical Application Component	Logical Application Component	Decomposes
Logical Application Component	Logical Application Component	Communicates with
Logical Application Component	Logical Data Component	Used by
Logical Application Component	Logical Technology Component	Is served by
Logical Application Component	Physical Application Component	Is realized by
Logical Data Component	Data Entity	Encapsulates
Logical Data Component	Logical Application Component	Uses
Logical Data Component	Physical Data Component	Is realized by
Logical Technology Component	Business Service	Provides platform for
Logical Technology Component	Logical Application Component	Serves
Logical Technology Component	Logical Technology Component	Decomposes
Logical Technology Component	Logical Technology Component	Is dependent on
Logical Technology Component	Physical Technology Component	Is realized by
Logical Technology Component	Technology Service	Supplies
Measure	Measure	Decomposes
Measure	Objective	Sets performance criteria for
Objective	Goal	Realizes
Objective	Measure	Is tracked against
Objective	Objective	Decomposes
Organization Unit	Actor	Contains
Organization Unit	Business Capability	Delivers
Organization Unit	Business Service	Owns and governs
Organization Unit	Course of Action	Participates in

Source Entity	Target Entity	Name
Organization Unit	Driver	Is motivated by
Organization Unit	Function	Enables
Organization Unit	Organization Unit	Decomposes
Organization Unit	Product	Delivers
Physical Application Component	Logical Application Component	Realizes
Physical Application Component	Physical Application Component	Decomposes
Physical Application Component	Physical Application Component	Communicates with
Physical Application Component	Physical Data Component	Used by
Physical Application Component	Physical Technology Component	Is served by
Physical Data Component	Logical Data Component	Realizes
Physical Data Component	Physical Application Component	Used by
Physical Data Component	Physical Data Component	Decomposes
Physical Technology Component	Logical Technology Component	Realizes
Physical Technology Component	Physical Application Component	Serves
Physical Technology Component	Physical Technology Component	Decomposes
Physical Technology Component	Physical Technology Component	Is dependent on
Process	Actor	Is produced by
Process	Actor	Supports
Process	Business Capability	Operationalizes
Process	Business Information	Is used by
Process	Business Service	Orchestrates
Process	Business Service	Decomposes
Process	Control	Is guided by
Process	Event	Generates
Process	Event	Resolves
Process	Function	Supports
Process	Function	Is realized by
Process	Process	Decomposes
Process	Process	Precedes, follows
Process	Product	Delivers
Process	Role	Involves
Process	Role	Is performed by
Process	Value Stream	Operationalizes
Product	Course of Action	Is produced by
Product	Organization Unit	Is produced by
Product	Process	Is produced by
Role	Actor	Is performed by
Role	Process	Participates in
Role	Process	Performs
Role	Role	Decomposes
Service Quality	Contract	Applies to
Service Quality	Service	Applies to
Technology Service	Application Service	Serves
Technology Service	Logical Technology Component	Is supplied by
Value Stream	Actor	Involves

Source Entity	Target Entity	Name
Value Stream	Actor	Is triggered by
Value Stream	Business Capability	Is enabled by
Value Stream	Course of Action	Is influenced by
Value Stream	Process	Is operationalized by
Work Package	Capability	Delivers

Chapter 3: Architectural Artifacts

This chapter discusses the concepts surrounding architecture artifacts and then describes the artifacts that are recommended to be created for each phase within the ADM.

3.1 Basic Concepts

Architectural artifacts are created in order to describe a system, solution, or state of the enterprise. The concepts discussed in this section have been adapted from more formal definitions contained in ISO/IEC/IEEE 42010:2011 and ISO/IEC/IEEE 15288:2015. They are illustrated in [Figure 3-1](#).

The "environment" of a system is the context determining the setting and circumstances of all influences upon a system. The environment of a system includes developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological, and social influences.

A "system" is a combination of interacting elements organized to achieve one or more stated purposes.

The "architecture" of a system is the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution.

An "Architecture Description" is a work product used to express an architecture; a collection of architecture views and models that together document the architecture.

"Stakeholders" are individuals, teams, organizations, or classes thereof, having an interest in a system.

"Concerns" are interests in a system relevant to one or more of its stakeholders. Concerns may pertain to any aspect of the system's functioning, development, or operation, including considerations such as performance, reliability, security, distribution, and evolvability and may determine the acceptability of the system.

An "architecture view" is a representation of a system from the perspective of a related set of concerns. It consists of one or more architecture models of the system.

An "Architecture Model" is a representation of a subject of interest. A model provides a smaller scale, simplified, and/or abstract representation of the subject matter.

In capturing or representing the design of a system architecture, the architect will typically create one or more architecture models, possibly using different tools. An architecture view will comprise selected parts of one or more models, chosen so as to demonstrate to a particular stakeholder or group of stakeholders that their concerns are being adequately addressed in the design of the system architecture.

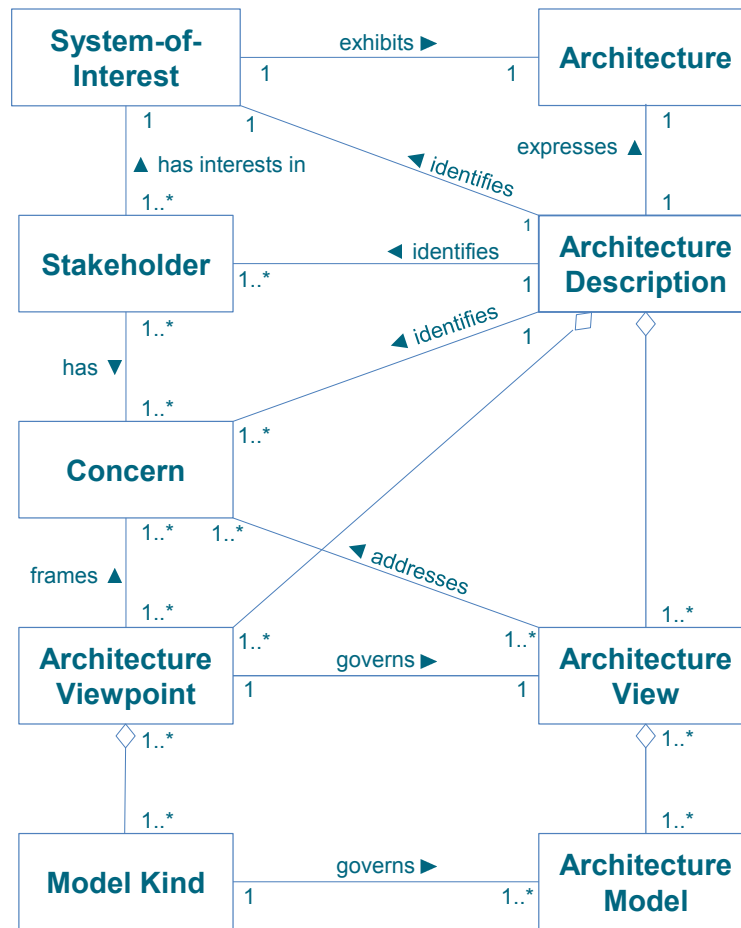


Figure 3-1 Basic Architectural Concepts

An "architecture viewpoint" is a specification of the conventions for a particular kind of architecture view. It can also be called the definition or schema for that kind of architecture view. It establishes the conventions for constructing, interpreting, and using an architecture view to address a specific concern (or set of concerns) about a system-of-interest.

A "Model Kind" establishes conventions for a type of modeling.

An architecture viewpoint references one or more model kinds; an architecture view incorporates one or more models.

A "viewpoint library" is a collection of the specifications of architecture viewpoints contained in the Reference Library portion of the Architecture Repository.

- An architecture view is what you see; an architecture viewpoint is where you are looking from — the vantage point or perspective that determines what you see

- Architecture viewpoints are generic, and can be stored in libraries for re-use; an architecture view is always specific to the architecture for which it is created
- Every architecture view has an associated architecture viewpoint that describes it, at least implicitly

ISO/IEC/IEEE 42010:2011 encourages architects to define architecture viewpoints explicitly. Making this distinction between the content and schema of a view may seem at first to be an unnecessary overhead, but it provides a mechanism for re-using architecture viewpoints across different architectures.

In summary, then, architecture views are representations of the overall architecture in terms meaningful to stakeholders. They enable the architecture to be communicated to and understood by the stakeholders, so they can verify that the system will address their concerns.

Concerns are often related to requirements. A concern can be a general requirement type, such as availability. It may lead to the definition of several particular requirements. It may be an interest related to some goal of a stakeholder. Identifying concerns helps ensure stakeholders' interests are addressed and requirements are identified. Associating concerns with general artifact types helps architects to select and develop particular artifacts for presentation to stakeholders.

3.1.1 Simple Example of an Architecture Viewpoint and Architecture View

For many architectures, a useful architecture viewpoint is that of business domains, which can be illustrated by an example from The Open Group itself.

The architecture viewpoint is specified as follows:

Architecture Viewpoint Element	Description
Stakeholders	Management Board, Chief Executive Officer
Concerns	Show the top-level relationships between US/UK 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.

The corresponding architecture view of The Open Group (in 2017) is shown in [Figure 3-2](#).

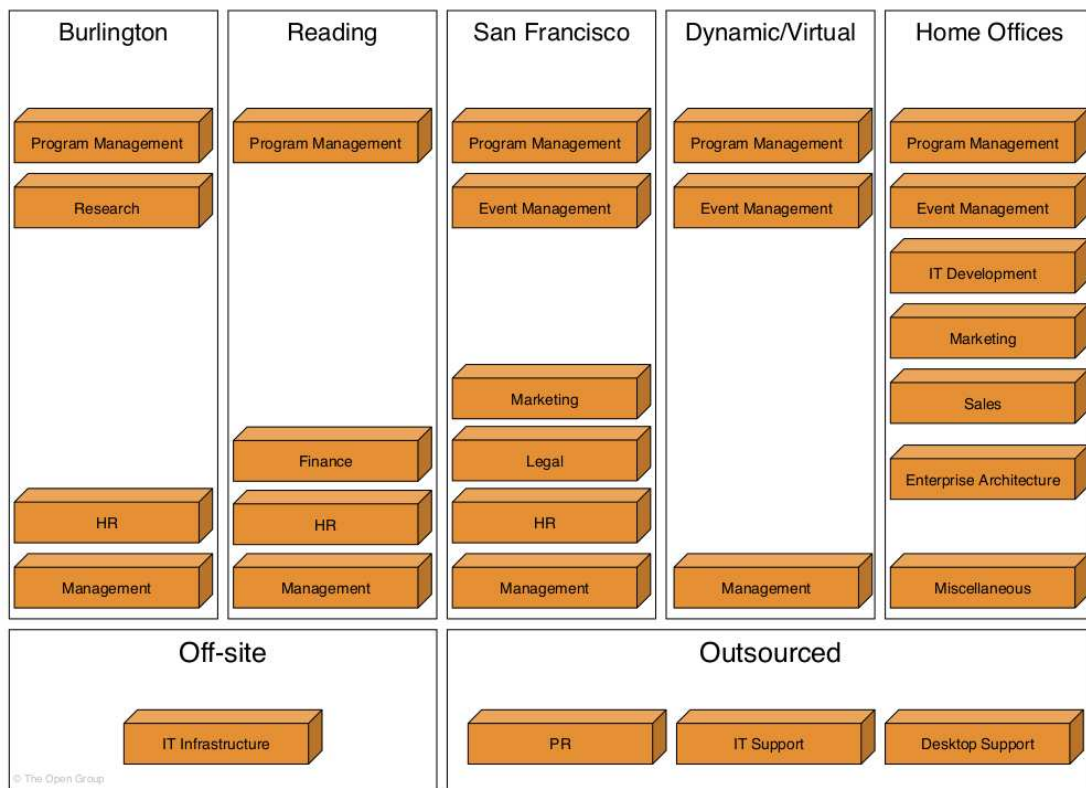


Figure 3-2 Example Architecture View — The Open Group Business Domains

3.2 Developing Architecture Views in the ADM

3.2.1 General Guidelines

The choice of which particular architecture views to develop is one of the key decisions that the architect has to make.

The architect has a responsibility for ensuring the completeness (fitness-for-purpose) of the architecture, in terms of adequately addressing all the pertinent concerns of its stakeholders; and the integrity of the architecture, in terms of connecting all the various views to each other, satisfactorily reconciling the conflicting concerns of different stakeholders, and showing the trade-offs made in so doing (as between security and performance, for example).

The choice has to be constrained by considerations of practicality, and by the principle of fitness-for-purpose (i.e., the architecture should be developed only to the point at which it is fit-for-purpose, and not reiterated *ad infinitum* as an academic exercise).

As explained in the TOGAF Standard — Architecture Development Method, the development of architecture views is an iterative process. The typical progression is from business to technology, using a technique such as business scenarios (see the TOGAF® Series Guide: Business

Scenarios) to properly identify all pertinent concerns; and from high-level overview to lower-level detail, continually referring back to the concerns and requirements of the stakeholders throughout the process.

Moreover, each of these progressions has to be made for two distinct environments: the existing environment (referred to as the baseline in the ADM) and the target environment. The architect must develop pertinent architecture views of both the Baseline Architecture and the Target Architecture. This provides the context for the gap analysis at the end of Phases B, C, and D of the ADM, which establishes the elements of the Baseline Architecture to be carried forward and the elements to be added, removed, or replaced.

This whole process is explained in the TOGAF Standard — ADM Techniques.

3.2.2 Architecture View Creation Process

As mentioned above, the TOGAF framework encourages but does not mandate the use of ISO/IEC/IEEE 42010:2011. The following description therefore covers both the situation where ISO/IEC/IEEE 42010:2011 has been adopted and where it has not.

ISO/IEC/IEEE 42010:2011 itself does not require any specific process for developing architecture viewpoints or creating views from them. Where ISO/IEC/IEEE 42010:2011 has been adopted and become well-established practice within an organization, it will often be possible to create the required views for a particular architecture by following these steps:

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

This approach can be expected to bring the following benefits:

- Less work for the architects (because the architecture viewpoints have already been defined and therefore the views can be created faster)
- Better comprehensibility for stakeholders (because the architecture viewpoints are already familiar)
- Greater confidence in the validity of the views (because their architecture viewpoints have a known track record)

However, situations can always arise in which an architecture view is needed for which no appropriate architecture viewpoint has been predefined. This is also the situation, of course, when an organization has not yet incorporated ISO/IEC/IEEE 42010:2011 into its architecture practice and established a library of architecture viewpoints.

In each case, the architect may choose to develop a new architecture viewpoint that will cover the outstanding need, and then generate an architecture view from it. (This is ISO/IEC/IEEE 42010:2011 recommended practice.) Alternatively, a more pragmatic approach can be equally successful: the architect can create an *ad hoc* architecture view for a specific system and later consider whether a generalized form of the implicit architecture viewpoint should be defined explicitly and saved in a library, so that it can be re-used. (This is one way of establishing a library of architecture viewpoints initially.)

Whatever the context, the architect should be aware that every architecture view has an

architecture viewpoint, at least implicitly, and that defining the architecture viewpoint in a systematic way (as recommended by ISO/IEC/IEEE 42010:2011) will help in assessing its effectiveness; i.e., does the architecture viewpoint cover the relevant stakeholder concerns?

3.3 Views, Tools, and Languages

The need for architecture views, and the process of developing them following the ADM, is explained above. This section describes the relationships between architecture views, the tools used to develop and analyze them, and a standard language enabling interoperability between the tools.

3.3.1 Overview

In order to achieve the goals of completeness and integrity in an architecture, architecture views are usually developed, visualized, communicated, and managed using a tool.

In the current state of the market, different tools normally have to be used to develop and analyze different views of the architecture. It is highly desirable that an Architecture Description be encoded in a standard language, to enable a standard approach to the description of architecture semantics and their re-use among different tools.

An architecture viewpoint is also normally developed, visualized, communicated, and managed using a tool, and it is also highly desirable that standard architecture viewpoints (i.e., templates or schemas) be developed, so that different tools that deal in the same views can interoperate, the fundamental elements of an architecture can be re-used, and the Architecture Description can be shared among tools.

Issues relating to the evaluation of tools for architecture work are discussed in detail in the TOGAF Standard — Architecture Development Method.

3.4 Architecture Views and Architecture Viewpoints

3.4.1 Example of Architecture Views and Architecture Viewpoints

To illustrate the concepts of architecture views and architecture viewpoints, consider the example of a very simple airport system with two different stakeholders: the pilot and the air traffic controller.

One architecture view can be developed from the architecture viewpoint of the pilot, which addresses the pilot's concerns. Equally, another architecture view can be developed from the architecture viewpoint of the air traffic controller. Neither architecture view completely describes the system in its entirety, because the architecture viewpoint of each stakeholder constrains (and reduces) how each sees the overall system.

The architecture viewpoint of the pilot comprises some concerns that are not relevant to the controller, such as passengers and fuel, while the architecture viewpoint of the controller comprises some concerns not relevant to the pilot, such as other planes. There are also elements shared between the two architecture viewpoints, such as the communication model between the pilot and the controller, and the vital information about the plane itself.

An architecture viewpoint is a model (or description) of the information contained in a view. In

our example, one architecture viewpoint is the description of how the pilot sees the system, and the other architecture viewpoint is how the controller sees the system.

Pilots describe the system from their perspective, using a model of their position and vector toward or away from the runway. All pilots use this model, and the model has a specific language that is used to capture information and populate the model.

Controllers describe the system differently, using a model of the airspace and the locations and vectors of aircraft within the airspace. Again, all controllers use a common language derived from the common model in order to capture and communicate information pertinent to their architecture viewpoint.

Fortunately, when controllers talk with pilots, they use a common communication language. (In other words, the models representing their individual architecture viewpoints partially intersect.) Part of this common language is about location and vectors of aircraft, and is essential to safety.

So, in essence, each architecture viewpoint is an abstract model of how all the stakeholders of a particular type — all pilots, or all controllers — view the airport system.

Tools exist to assist stakeholders, especially when they are interacting with complex models such as the model of an airspace, or the model of air flight.

The interface to the human user of a tool is typically close to the model and language associated with the architecture viewpoint. The unique tools of the pilot are fuel, altitude, speed, and location indicators. The main tool of the controller is radar. The common tool is a radio.

To summarize from the above example, we can see that an architecture view can subset the system through the perspective of the stakeholder, such as the pilot *versus* the controller. This subset can be described by an abstract model called an architecture viewpoint, such as an air flight *versus* an air space model. This description of the architecture view is documented in a partially specialized language, such as "pilot-speak" *versus* "controller-speak". Tools are used to assist the stakeholders, and they interface with each other in terms of the language derived from the architecture viewpoint ("pilot-speak" *versus* "controller-speak").

When stakeholders use common tools, such as the radio contact between pilot and controller, a common language is essential.

3.4.2 Architecture Views and Architecture Viewpoints in Enterprise Architecture

Now let us map this example to the Enterprise Architecture. Consider two stakeholders in a new small computing system: the users and the developers.

The users of the system have an architecture viewpoint that reflects their concerns when interacting with the system, and the developers of the system have a different architecture viewpoint. Architecture views that are developed to address either of the two architecture viewpoints are unlikely to exhaustively describe the whole system, because each perspective reduces how each sees the system.

The architecture viewpoint of the user is comprised of all the ways in which the user interacts with the system, not seeing any details such as applications or Database Management Systems (DBMS).

The architecture viewpoint of the developer is one of productivity and tools, and doesn't include things such as actual live data and connections with consumers.

However, there are things that are shared, such as descriptions of the processes that are

enabled by the system and/or communications protocols set up for users to communicate problems directly to development.

In this example, one architecture viewpoint is the description of how the user sees the system, and the other architecture viewpoint is how the developer sees the system. Users describe the system from their perspective, using a model of availability, response time, and access to information. All users of the system use this model, and the model has a specific language.

Developers describe the system differently than users, using a model of software connected to hardware distributed over a network, etc. However, there are many types of developers (database, security, etc.) of the system, and they do not have a common language derived from the model.

3.4.3 Need for a Common Language and Interoperable Tools for Architecture Description

Tools exist for both users and developers. Tools such as online help are there specifically for users, and attempt to use the language of the user. Many different tools exist for different types of developers, but they suffer from the lack of a common language that is required to bring the system together. It is difficult, if not impossible, in the current state of the tools market to have one tool interoperate with another tool.

Issues relating to the evaluation of tools for architecture work are discussed in detail in the TOGAF Standard — Architecture Development Method.

3.5 Conclusions

This section attempts to deal with views in a structured manner, but this is by no means a complete treatise on views.

In general, the TOGAF framework embraces the concepts and definitions presented in ISO/IEC/IEEE 42010:2011, specifically the concepts that help guide the development of an architecture view and make the architecture view actionable. These concepts can be summarized as:

- Selecting a key stakeholder
- Understanding their concerns and generalizing/documenting those concerns
- Understanding how to model and deal with those concerns

3.6 Architectural Artifacts by ADM Phase

Catalog, Matrix, and Diagram Concept

The Enterprise Metamodel is used as a technique to structure architectural information in an ordered way so that it can be processed to meet the stakeholder needs. The majority of architecture stakeholders do not actually need to know what the architecture metamodel is and are only concerned with specific issues, such as "what functionality does this application support?", "which processes will be impacted by this project?", etc. In order to meet the needs of these stakeholders, the TOGAF concepts of building blocks, catalogs, matrices, and diagrams are used.

Building blocks are entities of a particular type within the metamodel (for example, a business service called "Purchase Order"). Building blocks carry metadata according to the metamodel, which supports query and analysis. For example, business services have a metadata attribute for owner, which allows a stakeholder to query all business services owned by a particular organization. Building blocks may also include dependent or contained entities as appropriate to the context of the architecture (for example, a business service called "Purchase Order" may implicitly include a number of processes, data entities, application components, etc.).

Catalogs are lists of building blocks of a specific type, or of related types, that are used for governance or reference purposes (for example, an organization chart, showing locations and actors). As with building blocks, catalogs carry metadata according to the metamodel, which supports query and analysis.

Matrices are grids that show relationships between two or more model entities. Matrices are used to represent relationships that are list-based rather than graphical in their usage (for example, a CRUD matrix showing which applications Create, Read, Update, and Delete a particular type of data is difficult to represent visually).

Diagrams are renderings of architectural content in a graphical format to allow stakeholders to retrieve the required information. Diagrams can also be used as a technique for graphically populating architecture content or for checking the completeness of information that has been collected. The TOGAF Content Framework defines a set of architecture diagrams to be created (e.g., organization chart). Each of these diagrams may be created several times for an architecture with different style or content coverage to suit stakeholder concerns.

Building blocks, catalogs, matrices, and diagrams are all concepts that are well supported by leading Enterprise Architecture tools. In environments where tools are used to model the architecture, such tools typically support mechanisms to search, filter, and query the Architecture Repository.

On-demand querying of the Architecture Repository (such as the business service ownership example mentioned above) can be used to generate *ad hoc* catalogs, matrices, and diagrams of the architecture. As this type of query is by nature required to be flexible, it is therefore not restricted or defined within the Enterprise Metamodel.

The interactions between metamodel, building blocks, diagrams, and stakeholders are shown in [Figure 3-3](#). [Figure 3-4](#) shows the artifacts that are associated with the TOGAF Enterprise Metamodel.

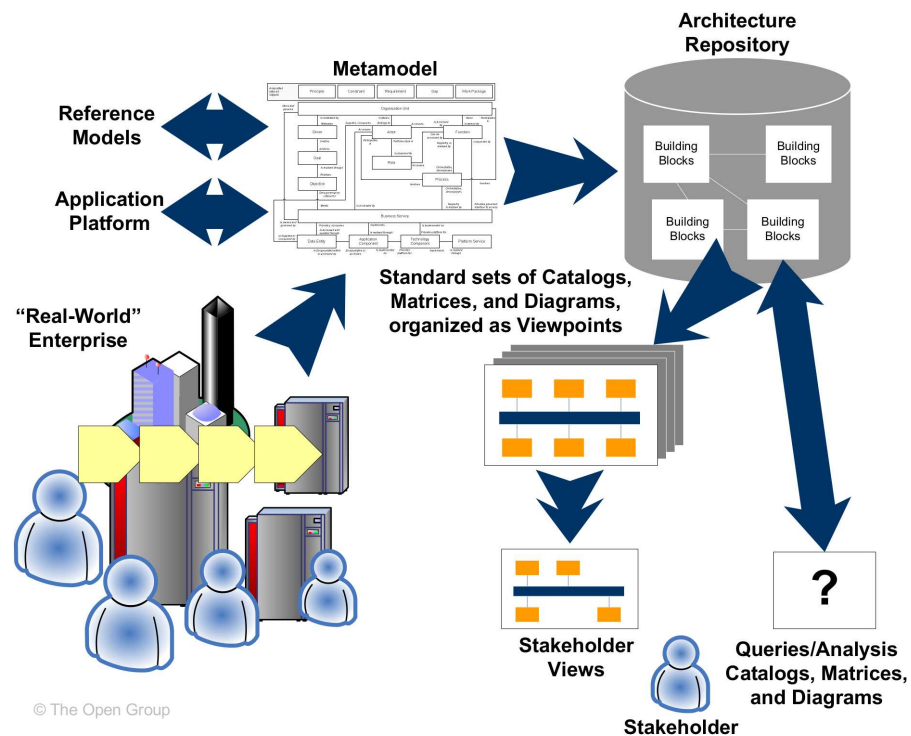


Figure 3-3 Interactions between Metamodel, Building Blocks, Diagrams, and Stakeholders

3.6.2 Phase A: Architecture Vision

The following describes catalogs, matrices, and diagrams that may be created within Phase A (Architecture Vision) as described in the TOGAF Standard — Architecture Development Method.

Stakeholder Catalog

The purpose of the Stakeholder catalog is to identify the stakeholders for the architecture engagement, their influence over the engagement, and their key questions, issues, or concerns that must be addressed by the architecture framework.

Understanding stakeholders and their requirements allows an architect to focus effort in areas that meet the needs of stakeholders (see the TOGAF Standard — ADM Techniques).

Due to the potentially sensitive nature of stakeholder mapping information and the fact that the Architecture Vision phase is intended to be conducted using informal modeling techniques, no specific metamodel entities will be used to generate a Stakeholder catalog.

Value Chain Diagram

A Value Chain diagram provides a high-level orientation view of an enterprise and how it interacts with the outside world. In contrast to the more formal organization map developed within Phase B (Business Architecture), the Value Chain diagram focuses on presentational impact.

The purpose of this diagram is to quickly on-board and align stakeholders for a particular change initiative, so that all participants understand the high-level functional and organizational context of the architecture engagement.

Solution Concept Diagram

A Solution Concept diagram provides a high-level orientation of the solution that is envisaged in order to meet the objectives of the architecture engagement. In contrast to the more formal and detailed architecture diagrams developed in the following phases, the solution concept represents a "pencil sketch" of the expected solution at the outset of the engagement.

This diagram may embody key objectives, requirements, and constraints for the engagement and also highlight work areas to be investigated in more detail with formal architecture modeling.

Its purpose is to quickly on-board and align stakeholders for a particular change initiative, so that all participants understand what the architecture engagement is seeking to achieve and how it is expected that a particular solution approach will meet the needs of the enterprise.

Business Model Diagram

A model describing the rationale for how an enterprise creates, delivers, and captures value.

Business Capability Map

A diagram that shows the business capabilities that an enterprise needs to meet its purposes.

Value Stream Map

A diagram representing an end-to-end collection of value-adding activities that create an overall result for a customer, stakeholder, or end user.

3.6.3 Phase B: Business Architecture

The following describes catalogs, matrices, and diagrams that may be created within Phase B (Business Architecture) as described in the TOGAF Standard — Architecture Development Method.

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 requirements in order to test for completeness.

For example, requirements for an application that services customers can be tested for completeness by verifying exactly which customer types need to be supported and whether there are any particular requirements or restrictions for user types.

The Organization/Actor catalog contains the following metamodel entities:

- Organization Unit
- Actor
- Location (may be included in this catalog if an independent Location catalog is not maintained)

Driver/Goal/Objective Catalog

The purpose of the Driver/Goal/Objective catalog is to provide a cross-organizational reference of how an organization meets its drivers in practical terms through goals, objectives, and (optionally) measures.

Publishing a definitive breakdown of drivers, goals, and objectives allows change initiatives within the enterprise to identify synergies across the organization (e.g., multiple organizations attempting to achieve similar objectives), which in turn allow stakeholders to be identified and related change initiatives to be aligned or consolidated.

The Driver/Goal/Objective catalog contains the following metamodel entities:

- Organization Unit
- Driver
- Goal
- Objective

- Measure (may optionally be included)

Role Catalog

The purpose of the Role catalog is to provide a listing of all authorization levels or zones within an enterprise. Frequently, application security or behavior is defined against locally understood concepts of authorization that create complex and unexpected consequences when combined on the user desktop.

If roles are defined, understood, and aligned across organizations and applications, this allows for a more seamless user experience and generally more secure applications, as administrators do not need to resort to workarounds in order to enable users to carry out their jobs.

In addition to supporting security definition for the enterprise, the Role catalog also forms a key input to identifying organizational change management impacts, defining job functions, and executing end-user training.

As each role implies access to a number of business functions, if any of these business functions are impacted then change management will be required, organizational responsibilities may need to be redefined, and retraining may be needed.

The Role catalog contains the following metamodel entities:

- Role

Business Service/Function Catalog

The purpose of the Business Service/Function catalog is to provide a functional decomposition in a form that can be filtered, reported on, and queried, as a supplement to graphical Functional Decomposition diagrams.

The Business Service/Function catalog can be used to identify capabilities of an organization and to understand the level that governance is applied to the functions of an organization. This functional decomposition can be used to identify new capabilities required to support business change or may be used to determine the scope of change initiatives, applications, or technology components.

The Business Service/Function catalog contains the following metamodel entities:

- Organization Unit
- Business Function
- Business Service
- Application Service (may optionally be included here)

Location Catalog

The Location catalog provides a listing of all locations where an enterprise carries out business operations or houses architecturally relevant assets, such as data centers or end-user computing equipment.

Maintaining a definitive list of locations allows change initiatives to quickly define a location scope and to test for completeness when assessing current landscapes or proposed target solutions. For example, a project to upgrade desktop operating systems will need to identify all locations where desktop operating systems are deployed.

Similarly, when new systems are being implemented a diagram of locations is essential in order

to develop appropriate deployment strategies that comprehend both user and application location and identify location-related issues, such as internationalization, localization, timezone impacts on availability, distance impacts on latency, network impacts on bandwidth, and access.

The Location catalog contains the following metamodel entities:

- Location

Process/Event/Control/Product Catalog

The Process/Event/Control/Product catalog provides a hierarchy of processes, events that trigger processes, outputs from processes, and controls applied to the execution of processes. This catalog provides a supplement to any Process Flow diagrams that are created and allows an enterprise to filter, report, and query across organizations and processes to identify scope, commonality, or impact.

For example, the Process/Event/Control/Product catalog allows an enterprise to see relationships of processes to sub-processes in order to identify the full chain of impacts resulting from changing a high-level process.

The Process/Event/Control/Product catalog contains the following metamodel entities:

- Process
- Event
- Control
- Product

Contract/Measure Catalog

The Contract/Measure catalog provides a listing of all agreed service contracts and the measures attached to those contracts. It forms the master list of service levels agreed to across the enterprise.

The Contract/Measure catalog contains the following metamodel entities:

- Business Service
- Application Service (optionally)
- Contract
- Measure

Business Capabilities Catalog

A listing of abilities that a business may possess to achieve specific purposes.

Value Stream Catalog

A listing of end-to-end collections of value-adding activities that create an overall result for a customer, stakeholder, or end user.

Value Stream Stages Catalog

A listing of end-to-end collections of the different stages for the value-adding activities that create an overall result for customer, stakeholder, or end user.

The Value Stream Stages catalog includes the following metamodel entities:

- Business Capability
- Value Stream

Business Glossary Catalog

This catalog contains the name and unambiguous description of all elements contained in or interacting with the Enterprise Architecture.

Optional information could include element enterprise synonyms, abbreviations/acronyms, and relationships with other elements. This catalog provides the semantics to be used by all analysts (e.g., business, information, and system) for their architecture products, including information/data models, and evolves throughout the ADM.

Business Interaction Matrix

The purpose of this matrix is to depict the relationship interactions between organizations and business functions across the enterprise.

Understanding business interaction of an enterprise is important as it helps to highlight value chain and dependencies across organizations.

The Business Interaction matrix shows the following metamodel entities and relationships:

- Organization
- Business Function
- Business Service
- Business Service *communicates with* Business Service relationships
- Business Service *is dependent on* Business Service relationships

Actor/Role Matrix

The purpose of this matrix is to show which actors perform which roles, supporting definition of security and skills requirements.

Understanding Actor-to-Role relationships is a key supporting tool in definition of training needs, user security settings, and organizational change management.

The Actor/Role matrix shows the following metamodel entities and relationships:

- Actor
- Role
- Actor *performs* Role relationships

Value Stream/Capability Matrix

The purpose of this matrix is to show the capabilities required to support each stage of a value stream.

Strategy/Capability Matrix

The purpose of this matrix is to show the capabilities required to support specific strategy statements.

Capability/Organization Matrix

The purpose of this matrix is to show the organization elements that implement each capability. The Capability/Organization matrix includes the following metamodel entities:

- Business Capability
- Value Stream
- Organization Unit

Business Footprint Diagram

A Business Footprint diagram describes the links between business goals, organizational units, business functions, and services, and maps these functions to the technical components delivering the required capability.

A Business Footprint diagram provides a clear traceability between a technical component and the business goal that it satisfies, while also demonstrating ownership of the services identified.

A Business Footprint diagram demonstrates only the key facts linking organization unit functions to delivery services and is utilized as a communication platform for senior-level (CxO) stakeholders.

Business Service/Information Diagram

The Business Service/Information diagram shows the information needed to support one or more business services. The Business Service/Information diagram shows what data is consumed by or produced by a business service and may also show the source of information.

The Business Service/Information diagram shows an initial representation of the information present within the architecture and therefore forms a basis for elaboration and refinement within Phase C (Data Architecture).

Functional Decomposition Diagram

The purpose of the Functional Decomposition diagram is to show on a single page the capabilities of an organization that are relevant to the consideration of an architecture. By examining the capabilities of an organization from a functional perspective, it is possible to quickly develop models of what the organization does without being dragged into extended debate on how the organization does it.

Once a basic Functional Decomposition diagram has been developed, it becomes possible to layer heat maps on top of this diagram to show scope and decisions. For example, the capabilities to be implemented in different phases of a change program.

Product Lifecycle Diagram

This diagram shows the possible state transitions of a business product, from its creation or receipt to its sale, disposal, or destruction.

The product may be a product of any kind (physical, software, or service).

Goal/Objective/Business Service Diagram

The purpose of a Goal/Objective/Business Service diagram is to define the ways in which a business service contributes to the achievement of a business vision or strategy.

Business services are associated with the drivers, goals, objectives, and measures that they support, allowing the enterprise to understand which business services contribute to similar aspects of business performance. The Goal/Objective/Business Service diagram also provides qualitative input on what constitutes high performance for a particular business service.

Business Use-Case Diagram

A Business Use-Case diagram displays the relationships between consumers and providers of business services. Business services are consumed by actors or other business services and the Business Use-Case diagram provides added richness in describing business capability by illustrating how and when that capability is used.

The purpose of the Business Use-Case diagram is to help to describe and validate the interaction between actors and their roles to processes and functions. As the architecture progresses, the use-case can evolve from the business level to include data, application, and technology details. Architectural business use-cases can also be re-used in systems design work.

Organization Decomposition Diagram

An Organization Decomposition diagram describes the links between actor, roles, and location within an organization tree.

An organization map should provide a chain of command of owners and decision-makers in the organization. Although it is not the intent of the Organization Decomposition diagram to link goal to organization, it should be possible to intuitively link the goals to the stakeholders from the Organization Decomposition diagram.

Process Flow Diagram

The purpose of the Process Flow diagram is to depict all models and mappings related to the process metamodel entity.

Process Flow diagrams show sequential flow of control between activities and may utilize swim-lane techniques to represent ownership and realization of process steps. For example, the application that supports a process step may be shown as a swim-lane.

In addition to showing a sequence of activity, process flows can also be used to detail the controls that apply to a process, the events that trigger or result from completion of a process, and also the products that are generated from process execution.

Process Flow diagrams are useful in elaborating the architecture with subject specialists, as they allow the specialist to describe "how the job is done" for a particular function. Through this process, each process step can become a more fine-grained function and can then in turn be elaborated as a process.

Business Event Diagram

The purpose of the Business Event diagram is to depict the relationship between events and process.

Certain events — such as arrival of certain information (e.g., customer submits sales order) or a certain point in time (e.g., end of fiscal quarter) — cause work and certain actions need to be undertaken within the business. These are often referred to as "business events" or simply "events" and are considered as triggers for a process. It is important to note that the event has to trigger a process and generate a business response or result.

Business Capability Map

A diagram that shows the business capabilities that an enterprise needs to meet its purposes.

Business capabilities may be decomposed into sub-capabilities.

Value Stream Map

A diagram representing an end-to-end collection of value-adding activities that create an overall result for a customer, stakeholder, or end user.

The value stream map includes the following metamodel entities:

- Business Capability
- Value Stream

Organization Map

A diagram showing the relationships between the primary entities that make up the enterprise, its partners, and stakeholders.

Information Map

A collection of information concepts and their relationships to one another. Information concepts, in effect, reflect the business vocabulary; e.g., client, account, or product. Mapping information in the Business Architecture starts with listing those elements that matter most to the business as well as how they are described in business terms.

3.6.4 Phase C: Data Architecture

The following describes catalogs, matrices, and diagrams that may be created within Phase C (Data Architecture) as described in the TOGAF Standard — Architecture Development Method.

Data Entity/Data Component Catalog

This catalog identifies and maintains a list of the data used across the enterprise, relating data entities to data components showing how data entities are structured.

Its purpose is to:

- Identify all data used in the enterprise
- Encourage effective data use

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

- Data Entity
- Logical Data Component
- 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 enterprise. 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:

- Assign ownership of data entities to organizations
- Understand the data and information exchange requirements business services
- Support the gap analysis and determine whether any data entities are missing and need to be created
- Define application of origin, application of record, and application of reference for data entities
- Enable development of data governance programs across the enterprise (establish data steward, develop data standards pertinent to the business function, etc.)

The Data Entity/Business Function matrix shows the following entities and relationships:

- Data Entity
- Business Function
- Data Entity relationship to owning Organization Unit

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.

Applications will create, read, update, and delete specific data entities that are associated with them. For example, a Customer Relationship Management (CRM) application will create, read, update, and delete customer entity information.

The data entities in a package/packaged services environment can be classified as master data, reference data, transactional data, content data, and historical data. Applications that operate on the data entities include transactional applications, information management applications, and business warehouse applications.

The mapping of the Application Component-Data Entity relationship is an important step as it enables the following to take place:

- Assign access of data to specific applications in the organization
- Understand the degree of data duplication within different applications, and the scale of the data lifecycle
- Understand where the same data is updated by different applications

- Support the gap analysis and determine whether any of the applications are missing and as a result need to be created

The Application/Data matrix is a two-dimensional table with Logical Application Component on one axis and Data Entity on the other axis.

Conceptual Data Diagram

The key purpose of the Conceptual Data diagram is to depict the relationships between critical data entities within the enterprise. This diagram is developed to address the concerns of business stakeholders.

Techniques used include:

- Entity relationship models
- Simplified UML class diagrams

Logical Data Diagram

The key purpose of the Logical Data diagram is to show logical views of the relationships between critical data entities within the enterprise. This diagram is developed to address the concerns of:

- Application developers
- Database designers

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. Moreover, by assigning business value to data, an indication of the business criticality of application components can be gained.

Additionally, the diagram may show data replication and application ownership of the master reference for data. In this instance, it can show two copies and the master-copy relationship between them. This diagram can include services; that is, services encapsulate data and they reside in an application, or services that reside on an application and access data encapsulated within the application.

Data Security Diagram

This diagram shows which data is accessed by which roles, organization units, and applications. It may be shown as a diagram or presented as a matrix.

Its purpose is to:

- Demonstrate compliance with data privacy laws and regulations
- Show security threats arising as a result of data access
- Show which parties outside of the organization have access to data
- Show security measures applied to data access

Data Migration Diagram

This diagram shows how data is extracted from baseline or source location(s) and loaded into target location(s). It may show where data is transformed and/or cleansed on the way. In Phase C of the ADM, the diagram is likely to be at an overview level. Later, it may be elaborated to show which source data items map to which target data items.

Data Lifecycle Diagram

The Data Lifecycle diagram is an essential part of managing business data throughout its lifecycle from conception until disposal within the constraints of the business process.

The data is considered as an entity in its own right, decoupled from business process and activity. Each change in state is represented on the diagram which may include the event or rules that trigger that change in state.

The separation of data from process allows common data requirements to be identified which enables resource sharing to be achieved more effectively.

3.6.5 Phase C: Application Architecture

The following describes catalogs, matrices, and diagrams that may be created within Phase C (Application Architecture) as described in the TOGAF Standard — Architecture Development Method.

Application Portfolio Catalog

The purpose of this catalog is to identify and maintain a list of all the applications in the enterprise. This list helps to define the horizontal scope of change initiatives that may impact particular kinds of applications. An agreed Application Portfolio allows a standard set of applications to be defined and governed.

The Application Portfolio catalog provides a foundation on which to base the remaining matrices and diagrams. It is typically the start point of the Application Architecture phase.

The Application Portfolio catalog contains the following metamodel entities:

- Application Service
- Logical Application Component
- Physical Application Component

Interface Catalog

The purpose of the Interface catalog is to scope and document the interfaces between applications to enable the overall dependencies between applications to be scoped as early as possible.

Applications will create, read, update, and delete data within other applications; this will be achieved by some kind of interface, whether via a batch file that is loaded periodically, a direct connection to another application's database, or via some form of API or web service.

The mapping of the Application Component-Application Component entity relationship is an important step as it enables the following to take place:

- Understand the degree of interaction between applications, identifying those that are central in terms of their dependencies on other applications
- Understand the number and types of interfaces between applications
- Understand the degree of duplication of interfaces between applications
- Identify the potential for simplification of interfaces when considering the target Application Portfolio
- Support the gap analysis and determine whether any of the applications are missing and as a result need to be created

The Interface catalog contains the following metamodel entities:

- Logical Application Component
- Physical Application Component
- Application *communicates with* application relationship

Application/Organization Matrix

The purpose of this matrix is to depict the relationship between applications and organizational units within the enterprise.

Business operations are performed by organizational units. Some of the operations and services performed by those organizational units will be supported by applications. The mapping of the Application Component-Organization Unit relationship is an important step as it enables the following to take place:

- Assign usage of applications to the organization units that perform business operations
- Understand the application support requirements of the business services and processes carried out by an organization unit
- Support the gap analysis and determine whether any of the applications are missing and as a result need to be created
- Define the application set used by a particular organization unit

The Application/Organization matrix is a two-dimensional table with Logical/Physical Application Component on one axis and Organization Unit on the other axis.

The relationship between these two entities is a composite of a number of metamodel relationships that need validating:

- Organization Units *own* Services
- Actors that *belong to* Organization Units *use* Services
- Services are *realized by* Logical/Physical Application Components

Role/Application Matrix

The purpose of the Role/Application matrix is to depict the relationship between applications and the business roles that use them within the enterprise.

People in an organization interact with applications. During this interaction, these people assume a specific role to perform a task; for example, product buyer.

The mapping of the Application Component-Role relationship is an important step as it enables the following to take place:

- Assign usage of applications to the specific roles in the organization
- Understand the application security requirements of the business services and processes supporting the function, and check these are in line with current policy
- Support the gap analysis and determine whether any of the applications are missing and as a result need to be created
- Define the application set used by a particular business role; essential in any move to role-based computing

The Role/Application matrix is a two-dimensional table with Logical Application Component on one axis and Role on the other axis.

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. Some of the business functions and services will be supported by applications. The mapping of the Application Component-Function relationship is an important step as it enables the following to take place:

- Assign usage of applications to the business functions that are supported by them
- Understand the application support requirements of the business services and processes carried out
- Support the gap analysis and determine whether any of the applications are missing and as a result need to be created
- Define the application set used by a particular business function

The Application/Function matrix is a two-dimensional table with Logical Application Component on one axis and Function on the other axis.

The relationship between these two entities is a composite of a number of metamodel relationships that need validating:

- Function *is bounded by* Service
- Services are *realized by* Logical/Physical Application Components

Application Interaction Matrix

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

The mapping of the application interactions shows in matrix form the equivalent of the Interface Catalog or an Application Communication diagram.

The Application Interaction matrix is a two-dimensional table with Application Service, Logical Application Component, and Physical Application Component on both the rows and the columns of the table.

The relationships depicted by this matrix include:

- Application Service *consumes* Application Service
- Logical Application Component *communicates with* Logical Application Component
- Physical Application Component *communicates with* Physical Application Component

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. Interfaces may be associated with data entities where appropriate. Applications may be associated with business services where appropriate. Communication should be logical and should only show intermediary technology where it is architecturally relevant.

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 distribution of where the host application is executed and/or delivered in thin client scenarios; the distribution of where applications are developed, tested, and released; etc.

Analysis can reveal opportunities for rationalization, as well as duplication and/or gaps.

The purpose of this diagram is to clearly depict the business locations from which business users typically interact with the applications, but also the hosting location of the application infrastructure.

The diagram enables:

- Identification of the number of package instances needed to sufficiently support the user population that may be spread out geographically
- Estimation of the number and type of user licenses for the package or other software
- Estimation of the level of support needed for the users and the location of the support center
- Selection of system management tools, structure, and management system required to support the enterprise users/customers/partners both locally and remotely
- Appropriate planning for the technological components of the business, namely server sizing and network bandwidth, etc.

- Performance considerations while implementing application and technology architecture solutions

Users typically interact with applications in a variety of ways; for example:

- To support the operations of the business day-to-day
- To participate in the execution of a business process
- To access information (look-up, read)
- To develop the application
- To administer and maintain the application

Application Use-Case Diagram

An Application Use-Case diagram displays the relationships between consumers and providers of application services. Application services are consumed by actors or other application services and the Application Use-Case diagram provides added richness in describing application functionality by illustrating how and when that functionality is used.

The purpose of the Application Use-Case diagram is to help to describe and validate the interaction between actors and their roles with applications. As the architecture progresses, the use-case can evolve from functional information to include technical realization detail.

Application use-cases can also be re-used in more detailed systems design work.

Enterprise Manageability Diagram

The Enterprise Manageability diagram shows how one or more applications interact with application and technology components that support the operational management of a solution.

This diagram is really a filter on the Application Communication diagram, specifically for enterprise management class software.

Analysis can reveal duplication and gaps, and opportunities in the IT service management operation of an organization.

Process/Application Realization Diagram

The purpose of the Process/Application Realization diagram is to clearly depict the sequence of events when multiple applications are involved in executing a business process.

It enhances the Application Communication diagram by augmenting it with any sequencing constraints, and hand-off points between batch and real-time processing.

It would identify complex sequences that could be simplified, and identify possible rationalization points in the architecture in order to provide more timely information to business users. It may also identify process efficiency improvements that may reduce interaction traffic between applications.

Software Engineering Diagram

The Software Engineering diagram breaks applications into packages, modules, services, and operations from a development perspective.

It enables more detailed impact analysis when planning migration stages, and analyzing opportunities and solutions.

It is ideal for application development teams and application management teams when managing complex development environments.

Application Migration Diagram

The Application Migration diagram identifies application migration from baseline to target application components. It enables a more accurate estimation of migration costs by showing precisely which applications and interfaces need to be mapped between migration stages.

It would identify temporary applications, staging areas, and the infrastructure required to support migrations (for example, parallel run environments, etc).

Software Distribution Diagram

The Software Distribution diagram shows how application software is structured and distributed across the estate. It is useful in systems upgrade or application consolidation projects.

This diagram shows how physical applications are distributed across physical technology and the location of that technology.

This enables a clear view of how the software is hosted, but also enables managed operations staff to understand how that application software is maintained once installed.

3.6.6 Phase D: Technology Architecture

The following section describes catalogs, matrices, and diagrams that may be created within Phase D (Technology Architecture) as described in the TOGAF Standard — Architecture Development Method.

Technology Standards Catalog

The Technology Standards catalog documents the agreed standards for technology across the enterprise covering technologies, and versions, the technology lifecycles, and the refresh cycles for the technology.

Depending upon the organization, this may also include location or business domain-specific standards information.

This catalog provides a snapshot of the enterprise standard technologies that are or can be deployed, and also helps identify the discrepancies across the enterprise.

The Technology Standards catalog contains the following metamodel entities:

- Technology Service
- Logical Technology Component

- Physical Technology Component

If technology standards are currently in place, apply these to the Technology Portfolio catalog to gain a baseline view of compliance with technology standards.

Technology Portfolio Catalog

The purpose of this catalog is to identify and maintain a list of all the technology in use across the enterprise, including hardware, infrastructure software, and application software. An agreed technology portfolio supports the lifecycle management of technology products and versions and also forms the basis for the definition of technology standards.

The Technology Portfolio catalog provides a foundation on which to base the remaining matrices and diagrams. It is typically the start point of the Technology Architecture phase.

Technology registries and repositories also provide input into this catalog from a baseline and target perspective.

Technologies in the catalog should be classified against the defined taxonomy in use in the enterprise, such as the TOGAF TRM — see the TOGAF® Series Guide: The TOGAF® Technical Reference Model (TRM) — adapted as necessary to fit the classification of technology products in use.

The Technology Portfolio catalog contains the following metamodel entities:

- Technology Service
- Logical Technology Component
- Physical Technology Component

Application/Technology Matrix

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

This matrix should be aligned with and complement one or more platform decomposition diagrams.

The Application/Technology matrix shows:

- Logical/Physical Application Components
- Services, Logical Technology Components, and Physical Technology Components
- Physical Technology Component *realizes* Physical Application Component relationships

Environments and Locations Diagram

The Environments and Locations diagram depicts which locations host which applications, identifies what technologies and/or applications are used at which locations, and finally identifies the locations from which business users typically interact with the applications.

This diagram should also show the existence and location of different deployment environments, including non-production environments, such as development and pre-production.

Platform Decomposition Diagram

The Platform Decomposition diagram depicts the technology platform that supports the operations of the Information Systems Architecture. The diagram covers all aspects of the infrastructure platform and provides an overview of the enterprise's technology platform. The diagram can be expanded to map the technology platform to appropriate application components within a specific functional or process area. This diagram may show specification details, such as product versions, number of CPUs, etc. or simply could be an informal "eye-chart" providing an overview of the technical environment.

The diagram should clearly show the enterprise applications. The technology platform for each application area can further be decomposed as follows:

- Hardware:
 - Logical Technology Components (with attributes)
 - Physical Technology Components (with attributes)
- Software:
 - Logical Technology Components (with attributes)
 - Physical Technology Components (with attributes)

Depending upon the scope of the Enterprise Architecture work, additional technology cross-platform information (e.g., communications, telco, and video information) may be addressed.

Processing Diagram

The Processing diagram focuses on deployable units of code/configuration and how these are deployed onto the technology platform. A deployment unit represents the grouping of business capability, service, or application components. The Processing diagram addresses the following:

- Which set of application components need to be grouped to form a deployment unit
- How one deployment unit connects/interacts with another (LAN, WAN, and the applicable protocols)
- How application configuration and usage patterns generate load or capacity requirements for different technology components

The organization and grouping of deployment units depends on separation concerns of the presentation, business logic, and data store layers and service-level requirements of the components. For example, the presentation layer deployment unit is grouped based on the following:

- Application components that provide user interface or user access functions
- Application components that are differentiated by location and user roles

There are several considerations to determine how application components are grouped together. Each deployment unit is made up of sub-units, such as:

- **Installation:** part that holds the executable code or package configuration (in case of packages)
- **Execution:** application component with its associated state at run time

- **Persistence:** data that represents the persistent state of the application component

Finally, these deployment units are deployed on either dedicated or shared technology components (workstation, web server, application server, or database server, etc.). It is important to note that technology processing can influence and have implications on the services definition and granularity.

Networked Computing/Hardware Diagram

Starting with the transformation to client-server systems from mainframes and later with the advent of e-Business and J2EE, large enterprises moved predominantly into a highly network-based distributed network computing environment with firewalls and demilitarized zones. Currently, most of the applications have a web front-end and, looking at the deployment architecture of these applications, it is very common to find three distinct layers in the network landscape; namely a web presentation layer, a business logic or application layer, and a back-end data store layer. It is common practice for applications to be deployed and hosted in a shared and common infrastructure environment.

So it becomes highly critical to document the mapping between logical applications and the technology components (e.g., server) that supports the application both in the development and production environments. The purpose of this diagram is to show the "as deployed" logical view of logical application components in a distributed network computing environment. The diagram is useful for the following reasons:

- Enable understanding of which application is deployed where in the distributed network computing environment
- Establishing authorization, security, and access to these technology components
- Understand the Technology Architecture that supports the applications during problem resolution and troubleshooting
- Isolate performance problems encountered by applications, determine whether it is application code-related or technology platform-related, and perform necessary upgrade to specific physical technology components
- Identify areas of optimization as and when newer technologies are available which will eventually reduce cost
- Enable application/technology auditing and prove compliance with enterprise technology standards
- Serve as an important tool to introduce changes to the Technology Architecture, thereby supporting effective change management
- Establish traceability and changing application end-point address while moving application either from a shared environment to a dedicated environment or *vice versa*

The scope of the diagram can be appropriately defined to cover a specific application, business function, or the entire enterprise. If chosen to be developed at the enterprise level, then the network computing landscape can be depicted in an application-agnostic way as well.

Network and Communications Diagram

The Network and Communications diagram describes the means of communication — the method of sending and receiving information — between these assets in the Technology Architecture; insofar as the selection of package solutions in the preceding architectures put specific requirements on the communications between the applications.

The Network and Communications diagram will take logical connections between client and server components and identify network boundaries and network infrastructure required to physically implement those connections. It does not describe the information format or content, but will address protocol and capacity issues.

3.6.7 Phase E: Opportunities and Solutions

The following section describes catalogs, matrices, and diagrams that may be created within Phase E (Opportunities & Solutions) as described in the TOGAF Standard — Architecture Development Method.

Project Context Diagram

A Project Context diagram shows the scope of a work package to be implemented as a part of a broader transformation roadmap. The Project Context diagram links a work package to the organizations, functions, services, processes, applications, data, and technology that will be added, removed, or impacted by the project.

The Project Context diagram is also a valuable tool for project portfolio management and project mobilization.

Benefits Diagram

The Benefits diagram shows opportunities identified in an architecture definition, classified according to their relative size, benefit, and complexity. This diagram can be used by stakeholders to make selection, prioritization, and sequencing decisions on identified opportunities.

3.6.8 Requirements Management

The following section describes catalogs, matrices, and diagrams that may be created within the Requirements Management phase as described in the TOGAF Standard — Architecture Development Method.

Requirements Catalog

The Requirements catalog captures things that the enterprise needs to do to meet its objectives. Requirements generated from architecture engagements are typically implemented through change initiatives identified and scoped during Phase E (Opportunities & Solutions). Requirements can also be used as a quality assurance tool to ensure that a particular architecture is fit-for-purpose (i.e., the architecture can meet all identified requirements).

The Requirements catalog contains the following metamodel entities:

- Requirement
- Assumption
- Constraint
- Gap

Chapter 4: Architecture Deliverables

This chapter provides descriptions of deliverables referenced in the ADM.

4.1 Introduction

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

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

The TOGAF Content Framework (see [Chapter 1](#)) identifies deliverables that are produced as outputs from executing the ADM cycle and potentially consumed as inputs at other points in the ADM. Other deliverables may be produced elsewhere and consumed by the ADM.

Deliverables produced by executing the ADM are shown in the table below.

Deliverable	Output from...	Input to...
Architecture Building Blocks (see Section 4.2.1)	F, H	A, B, C, D, E
Architecture Contract (see Section 4.2.2)	G	G, H
Architecture Definition Document (see Section 4.2.3)	A, B, C, D, E, F	B, C, D, E, F, G, H
Architecture Principles (see Section 4.2.4)	Preliminary, A, B, C, D	Preliminary, A, B, C, D, E, F, G, H
Architecture Repository (see Section 4.2.5)	Preliminary	Preliminary, A, B, C, D, E, F, G, H, Requirements Management
Architecture Requirements Specification (see Section 4.2.6)	B, C, D, E, F, Requirements Management	C, D, Requirements Management
Architecture Roadmap (see Section 4.2.7)	B, C, D, E, F	B, C, D, E, F
Architecture Vision (see Section 4.2.8)	A, E	B, C, D, E, F, G, H, Requirements Management
Business Principles, Business Goals, and Business Drivers	Preliminary, A, B	A, B

Deliverable	Output from...	Input to...
(see Section 4.2.9)		
Capability Assessment (see Section 4.2.10)	A, E	B, C, D, E, F
Change Request (see Section 4.2.11)	F, G, H	—
Communications Plan (see Section 4.2.12)	A	B, C, D, E, F
Compliance Assessment (see Section 4.2.13)	G	H
Implementation and Migration Plan (see Section 4.2.14)	E, F	F
Implementation Governance Model (see Section 4.2.15)	F	G, H
Organizational Model for Enterprise Architecture (see Section 4.2.16)	Preliminary	Preliminary, A, B, C, D, E, F, G, H, Requirements Management
Request for Architecture Work (see Section 4.2.17)	Preliminary, F, H	A, G
Requirements Impact Assessment (see Section 4.2.18)	Requirements Management	Requirements Management
Solution Building Blocks (see Section 4.2.19)	G	A, B, C, D, E, F, G
Statement of Architecture Work (see Section 4.2.20)	A, B, C, D, E, F, G, H	B, C, D, E, F, G, H, Requirements Management
Tailored Architecture Framework (see Section 4.2.21)	Preliminary, A	Preliminary, A, B, C, D, E, F, G, H, Requirements Management

4.2 Deliverable Descriptions

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

Note that not all the content described here need be contained in a particular deliverable. Rather, it is recommended that external references be used where possible; for example, the strategic plans of a business should not be copied into a Request for Architecture Work, but rather the title of the strategic plans should be referenced.

Also, it is not suggested that these descriptions should be followed to the letter. However, each element should be considered carefully; ignoring any input or output item may cause problems downstream.

4.2.1 Architecture Building Blocks

Architecture documentation and models from the enterprise's Architecture Repository; see [Chapter 5](#).

4.2.2 Architecture Contract

Purpose

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

- A system of continuous monitoring to check integrity, changes, decision-making, and audit of all architecture-related activities within the organization
- Adherence to the principles, standards, and requirements of the existing or developing architectures
- Identification of risks in all aspects of the development and implementation of the architecture(s) covering the internal development against accepted standards, policies, technologies, and products as well as the operational aspects of the architectures such that the organization can continue its business within a resilient environment
- A set of processes and practices that ensure accountability, responsibility, and discipline with regard to the development and usage of all architectural artifacts
- A formal understanding of the governance organization responsible for the contract, their level of authority, and scope of the architecture under the governance of this body

Content

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
- Conformance requirements
- Architecture development and management process and roles
- Target Architecture measures
- Defined phases of deliverables
- Prioritized joint workplan
- Time window(s)
- Architecture delivery and business metrics

Typical contents of a Business Users' Architecture Contract are:

- Introduction and background
- The nature of the agreement
- Scope
- Strategic requirements
- Conformance requirements
- Architecture adopters
- Time window
- Architecture business metrics
- Service architecture (includes Service-Level Agreement (SLA))

For more detail on the use of Architecture Contracts, see the TOGAF Standard — Enterprise Architecture Capability and Governance.

4.2.3 Architecture Definition Document

Purpose

The Architecture Definition Document is the deliverable container for the core architectural artifacts created during a project and for important related information. The Architecture Definition Document spans all architecture domains (Business, Data, Application, and Technology) and also examines all relevant states of the architecture (Baseline, Transition, and Target).

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

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

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

Content

Typical contents of an Architecture Definition Document are:

- Scope
- Goals, objectives, and constraints
- Architecture Principles
- Baseline Architecture

- Architecture models (for each state to be modeled):
 - Business Architecture models
 - Data Architecture models
 - Application Architecture models
 - Technology Architecture models
- Rationale and justification for architectural approach
- Mapping to Architecture Repository:
 - Mapping to Architecture Landscape
 - Mapping to reference models
 - Mapping to standards
 - Re-use assessment
- Gap analysis
- Impact assessment
- Transition Architecture:
 - Definition of transition states
 - Business Architecture for each transition state
 - Data Architecture for each transition state
 - Application Architecture for each transition state
 - Technology Architecture for each transition state

4.2.4 Architecture Principles

Purpose

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

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

Content

See the TOGAF Standard — ADM Techniques for guidelines and a detailed set of generic Architecture Principles, including:

- Business principles (see the TOGAF Standard — ADM Techniques)
- Data principles (see the TOGAF Standard — ADM Techniques)
- Application principles (see the TOGAF Standard — ADM Techniques)
- Technology principles (see the TOGAF Standard — ADM Techniques)

4.2.5 Architecture Repository

Purpose

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

Content

See [Chapter 7](#) for a detailed description of the content of an Architecture Repository.

4.2.6 Architecture Requirements Specification

Purpose

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

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

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

Content

Typical contents of an Architecture Requirements Specification are:

- Success measures
- Architecture requirements
- Business service contracts
- Application service contracts
- Implementation guidelines
- Implementation specifications
- Implementation standards
- Interoperability requirements
- IT Service Management requirements
- Constraints
- Assumptions

4.2.7 Architecture Roadmap

Purpose

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

Content

Typical contents of an Architecture Roadmap are:

- Work package portfolio:
 - Work package description (name, description, objectives, deliverables)
 - Functional requirements
 - Dependencies
 - Relationship to opportunity
 - Relationship to Architecture Definition Document and Architecture Requirements Specification
 - Business value
- Implementation Factor catalog, including:
 - Risks
 - Issues
 - Assumptions
 - Dependencies
 - Actions
 - Inputs
- Consolidated Gaps, Solutions, and Dependencies matrix, including:
 - Architecture domain
 - Gap
 - Potential solutions
 - Dependencies
- Any Transition Architectures
- Implementation recommendations:
 - Criteria measures of effectiveness of projects

- Risks and issues
- Solution Building Blocks

4.2.8 Architecture Vision

Purpose

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

Content

Typical contents of an Architecture Vision are:

- Problem description:
 - Stakeholders and their concerns
 - List of issues/scenarios to be addressed
- Objective of the Statement of Architecture Work
- Summary views necessary for the Request for Architecture Work and the Draft Business, Data, Application, and Technology Architectures created; typically including:
 - Value Chain diagram
 - Solution Concept diagram
- Mapped requirements
- Reference to Draft Architecture Definition Document

4.2.9 Business Principles, Business Goals, and Business Drivers

Purpose

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

Content

The content and structure of business context for architecture is likely to vary considerably from one organization to the next.

4.2.10 Capability Assessment

Purpose

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

- What is the capability level of the enterprise as a whole? Where does the enterprise wish to increase or optimize capability? What are the architectural focus areas that will support the desired development of the enterprise?
- What is the capability or maturity level of the IT function within the enterprise? What are the likely implications of conducting the architecture project in terms of design governance, operational governance, skills, and organization structure? What is an appropriate style, level of formality, and amount of detail for the architecture project to fit with the culture and capability of the IT organization?
- What is the capability and maturity of the architecture function within the enterprise? What architectural assets are currently in existence? Are they maintained and accurate? What standards and reference models need to be considered? Are there likely to be opportunities to create re-usable assets during the architecture project?
- Where capability gaps exist, to what extent is the business ready to transform in order to reach the target capability? What are the risks to transformation, cultural barriers, and other considerations to be addressed beyond the basic capability gap?

Content

Typical contents of a Capability Assessment are:

- Business Capability Assessment, including:
 - Capabilities of the business
 - Baseline state assessment of the performance level of each capability
 - Future state aspiration for the performance level of each capability
 - Baseline state assessment of how each capability is realized
 - Future state aspiration for how each capability should be realized
 - Assessment of likely impacts to the business organization resulting from the successful deployment of the Target Architecture
- IT Capability Assessment, including:
 - Baseline and target maturity level of change process
 - Baseline and target maturity level of operational processes
 - Baseline capability and capacity assessment
 - Assessment of the likely impacts to the IT organization resulting from the successful deployment of the Target Architecture

- Architecture maturity assessment, including:
 - Architecture Governance processes, organization, roles, and responsibilities
 - Architecture skills assessment
 - Breadth, depth, and quality of landscape definition with the Architecture Repository
 - Breadth, depth, and quality of standards definition with the Architecture Repository
 - Breadth, depth, and quality of reference model definition with the Architecture Repository
 - Assessment of re-use potential
- Business Transformation Readiness Assessment, including:
 - Readiness factors
 - Vision for each readiness factor
 - Current and target readiness ratings
 - Readiness risks

4.2.11 Change Request

Purpose

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

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

Content

Typical contents of a Change Request are:

- Description of the proposed change
- Rationale for the proposed change
- Impact assessment of the proposed change, including:
 - Reference to specific requirements
 - Stakeholder priority of the requirements to date
 - Phases to be revisited
 - Phase to lead on requirements prioritization
 - Results of phase investigations and revised priorities

- Recommendations on management of requirements
- Repository reference number

4.2.12 Communications Plan

Purpose

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

Content

Typical contents of a Communications Plan are:

- Identification of stakeholders and grouping by communication requirements
- Identification of communication needs, key messages in relation to the Architecture Vision, communication risks, and CSFs
- Identification of mechanisms that will be used to communicate with stakeholders and allow access to architecture information, such as meetings, newsletters, repositories, etc.
- Identification of a communications timetable, showing which communications will occur with which stakeholder groups at what time and in what location

4.2.13 Compliance Assessment

Purpose

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. Periodic compliance reviews of implementation projects provide a mechanism to review project progress and ensure that the design and implementation is proceeding in line with the strategic and architectural objectives.

Content

Typical contents of a Compliance Assessment are:

- Overview of project progress and status
- Overview of project architecture/design
- Completed architecture checklists:
 - Hardware and operating system checklist
 - Software services and middleware checklist
 - Applications checklists

- Information management checklists
- Security checklists
- System management checklists
- System engineering checklists
- Methods and tools checklists

4.2.14 Implementation and Migration Plan

Purpose

The Implementation and Migration Plan provides a schedule of the projects that will realize the Target Architecture. The Implementation and Migration Plan includes executable projects grouped into managed portfolios and programs. The Implementation and Migration Strategy identifying the approach to change is a key element of the Implementation and Migration Plan.

Content

Typical contents of an Implementation and Migration Plan are:

- Implementation and Migration Strategy:
 - Strategic implementation direction
 - Implementation sequencing approach
- Project and portfolio breakdown of implementation:
 - Allocation of work packages to project and portfolio
 - Capabilities delivered by projects
 - Milestones and timing
 - Work breakdown structure
 - May include impact on existing portfolio, program, and projects

It may contain:

- Project charters:
 - Included work packages
 - Business value
 - Risk, issues, assumptions, dependencies
 - Resource requirements and costs
 - Benefits of migration, determined (including mapping to business requirements)
 - Estimated costs of migration options

4.2.15 Implementation Governance Model

Purpose

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

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

Content

Typical contents of an Implementation Governance Model are:

- Governance processes
- Governance organization structure
- Governance roles and responsibilities
- Governance checkpoints and success/failure criteria

4.2.16 Organizational Model for Enterprise Architecture

Purpose

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

Content

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)
- Constraints on architecture work
- Budget requirements
- Governance and support strategy

4.2.17 Request for Architecture Work

Purpose

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

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

Content

Requests for Architecture Work typically include:

- Organization sponsors
- Organization's mission statement
- Business goals (and changes)
- Strategic plans of the business
- Time limits
- Changes in the business environment
- Organizational constraints
- Budget information, financial constraints
- External constraints, business constraints
- Current business system description
- Current architecture/IT system description
- Description of developing organization
- Description of resources available to developing organization

4.2.18 Requirements Impact Assessment

Purpose

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

Content

Typical contents of a Requirements Impact Assessment are:

- Reference to specific requirements
- Stakeholder priority of the requirements to date
- Phases to be revisited

- Phase to lead on requirements prioritization
- Results of phase investigations and revised priorities
- Recommendations on management of requirements
- Repository reference number

4.2.19 Solution Building Blocks

Implementation-specific building blocks from the enterprise's Architecture Repository; see [Chapter 5](#).

4.2.20 Statement of Architecture Work

Purpose

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

Content

Typical contents of a Statement of Architecture Work are:

- Title
- Architecture project request and background
- Architecture project description and scope
- Overview of Architecture Vision
- Specific change of scope procedures
- Roles, responsibilities, and deliverables
- Acceptance criteria and procedures
- Architecture project plan and schedule
- Approvals

4.2.21 Tailored Architecture Framework

Purpose

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

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

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

See the TOGAF Standard — Architecture Development Method for further considerations when selecting and tailoring the architecture framework.

Content

Typical contents of a Tailored Architecture Framework are:

- Tailored architecture method
- Tailored architecture content (deliverables and artifacts)
- Configured and deployed tools
- Interfaces with governance models and other frameworks:
 - Corporate Business Planning
 - Enterprise Architecture
 - Portfolio, Program, Project Management
 - System Development/Engineering
 - Operations (Services)

Chapter 5: Building Blocks

This chapter explains the concept of building blocks.

5.1 Overview

This section is intended to explain and illustrate the concept of building blocks in architecture.

Following this overview, there are two main parts:

- Introduction to Building Blocks (see [Section 5.2](#)), discusses the general concepts of building blocks, and explains the differences between ABBs and SBBs
- Building Blocks and the ADM (see [Section 5.3](#)), summarizes the stages at which building block design and specification occurs within the TOGAF ADM

5.2 Introduction to Building Blocks

This section is an introduction to the concept of building blocks.

5.2.1 Overview

This section describes the characteristics of building blocks. The use of building blocks in the ADM is described separately in [Section 5.3](#).

5.2.2 Generic Characteristics

Building blocks have generic characteristics as follows:

- A building block is a package of functionality defined to meet the business needs across an organization
- A building block normally has a type that corresponds to the metamodel (such as actor, business service, application, or data entity)
- A building block has a defined boundary and is generally recognizable as "a thing" by domain experts
- A building block may interoperate with other, inter-dependent building blocks.
- A good building block has the following characteristics:
 - It considers implementation and usage, and evolves to exploit technology and standards

- It may be assembled from other building blocks
- It may be a subassembly of other building blocks
- Ideally a building block is re-usable and replaceable, and well specified

A building block's boundary and specification should be loosely coupled to its implementation; i.e., it should be possible to realize a building block in several different ways without impacting the boundary or specification of the building block. The way in which assets and capabilities are assembled into building blocks will vary widely between individual architectures. Every organization must decide for itself what arrangement of building blocks works best for it. A good choice of building blocks can lead to improvements in legacy system integration, interoperability, and flexibility in the creation of new systems and applications.

Systems are built up from collections of building blocks, so most building blocks have to interoperate with other building blocks. Wherever that is true, it is important that the interfaces to a building block are published and reasonably stable.

Building blocks can be defined at various levels of detail, depending on what stage of architecture development has been reached.

For instance, at an early stage, a building block can simply consist of a name or an outline description. Later on, a building block may be decomposed into multiple supporting building blocks and may be accompanied by a full specification.

The level of detail to which a building block should be specified is dependent on the objectives of the architecture and, in some cases, less detail may be of greater value (for example, when presenting the capabilities of an enterprise, a single clear and concise picture has more value than a dense 100-page specification).

The Object Management Group® (OMG®) has developed a standard for Re-usable Asset Specification (RAS),¹ which provides a good example of how building blocks can be formally described and managed.

5.2.3 Architecture Building Blocks

Architecture Building Blocks (ABBs) relate to the Architecture Continuum (see [Section 6.4.1](#)), and are defined or selected as a result of the application of the ADM.

5.2.3.1 Characteristics

ABBs:

- Capture architecture requirements; e.g., Business, Data, Application, and Technology requirements
- Direct and guide the development of SBBs

1. Refer to www.omg.org/spec/RAS.

5.2.3.2 *Specification Content*

ABB specifications include the following as a minimum:

- Fundamental functionality and attributes: semantic, unambiguous, including security capability and manageability
- Interfaces: chosen set, supplied
- Interoperability and relationship with other building blocks
- Dependent building blocks with required functionality and named user interfaces
- Map to business/organizational entities and policies

5.2.4 **Solution Building Blocks**

Solution Building Blocks (SBBs) relate to the Solutions Continuum (see [Section 6.4.2](#)), and may be either procured or developed.

5.2.4.1 *Characteristics*

SBBs:

- Define what products and components will implement the functionality
- Define the implementation
- Fulfil business requirements
- Are product or vendor-aware

5.2.4.2 *Specification Content*

SBB specifications include the following as a minimum:

- Specific functionality and attributes
- Interfaces; the implemented set
- Required SBBs used with required functionality and names of the interfaces used
- Mapping from the SBBs to the IT topology and operational policies
- Specifications of attributes shared across the environment (not to be confused with functionality) such as security, manageability, localizability, scalability
- Performance, configurability
- Design drivers and constraints, including the physical architecture
- Relationships between SBBs and ABBs

5.3 Building Blocks and the ADM

5.3.1 Basic Principles

This section focuses on the use of building blocks in the ADM. General considerations and characteristics of building blocks are described in [Section 5.2](#).

5.3.1.1 Building Blocks in Architecture Design

An architecture is a set of building blocks depicted in an architectural model, and a specification of how those building blocks are connected to meet the overall requirements of the business.

The various building blocks in an architecture specify the scope and approach that will be used to address a specific business problem.

There are some general principles underlying the use of building blocks in the design of specific architectures:

- An architecture need only contain building blocks that are relevant to the business problem that the architecture is attempting to address
- Building blocks may have complex relationships to one another

One building block may support multiple building blocks or may partially support a single building block (for example, the business service of "complaint handling" would be supported by many data entities and possibly multiple application components)
- Building blocks should conform to standards relevant to their type, the principles of the enterprise, and the standards of the enterprise

5.3.1.2 Building Block Design

The process of identifying building blocks includes looking for collections of capabilities or assets that interact with one another and then drawing them together or making them different:

- Consider three classes of building blocks:
 - Re-usable building blocks, such as legacy items
 - Building blocks to be the subject of development, such as new applications
 - Building blocks to be the subject of purchase; i.e., Commercial Off-The-Shelf (COTS) applications
- Use the desired level of integration to bind or combine functions into building blocks; for instance, legacy elements could be treated as large building blocks to avoid breaking them apart

In the early stages and during views of the highest-level enterprise, the building blocks are often kept at a broad integration definition. It is during these exercises that the services definitions can often be best viewed. As implementation considerations are addressed, more detailed views of building blocks can often be used to address implementation decisions, focus on the critical strategic decisions, or aid in assessing the value and future impact of commonality and re-usability.

5.3.2 Building Block Specification Process in the ADM

The process of building block definition takes place gradually as the ADM is followed, mainly in Phases A, B, C, and D. It is an evolutionary and iterative process because as definition proceeds, detailed information about the functionality required, the constraints imposed on the architecture, and the availability of products may affect the choice and the content of building blocks.

The key phases and steps of the ADM at which building blocks are evolved and specified are summarized in [Figure 5-1](#). The major work in these steps consists of identifying the ABBs required to meet the business goals and objectives. The selected set of ABBs is then refined in an iterative process to arrive at a set of SBBs which can either be bought off-the-shelf or custom developed.

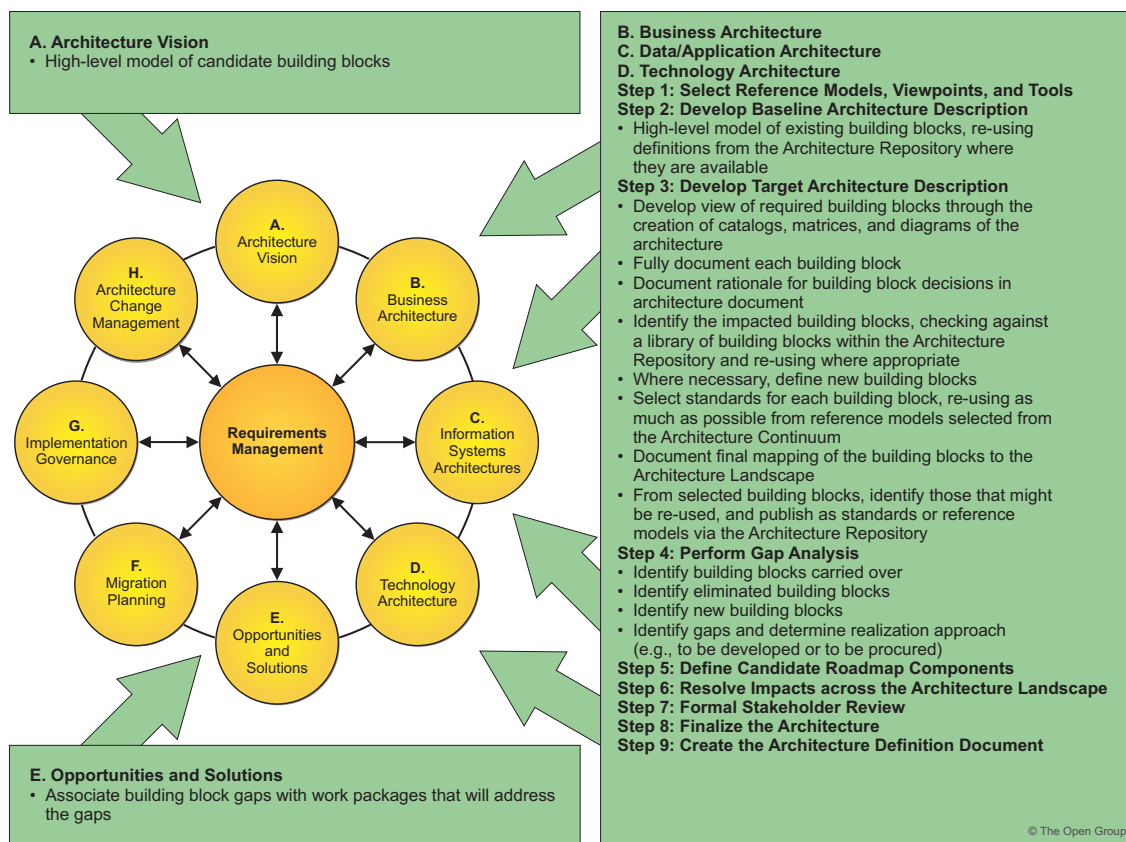


Figure 5-1 Key ADM Phases/Steps at which Building Blocks are Evolved/Specified

Chapter 6: Enterprise Continuum

6.1 Overview

The Enterprise Continuum provides methods for classifying architecture and solution artifacts, both internal and external to the Architecture Repository, as they evolve from generic Foundation Architectures to Organization-Specific Architectures.

The Enterprise Continuum enables the architect to articulate the broad perspective of what, why, and how the Enterprise Architecture has been designed with the factors and drivers considered. The Enterprise Continuum is an important aid to communication and understanding, both within individual enterprises, and between customer enterprises and vendor organizations. Without an understanding of "where in the continuum you are", people discussing architecture can often talk at cross-purposes because they are referencing different points in the continuum at the same time, without realizing it.

Any architecture is context-specific; for example, there are architectures that are specific to individual customers, industries, subsystems, products, and services. Architects, on both the buy side and supply side, must have at their disposal a consistent language to effectively communicate the differences between architectures. Such a language will enable engineering efficiency and the effective leveraging of COTS product functionality. The Enterprise Continuum provides that consistent language.

The Enterprise Continuum enables the organization of re-usable architecture artifacts and solution assets to maximize the Enterprise Architecture investment opportunities.

6.2 Enterprise Continuum and Architecture Re-Use

The simplest way of thinking of the Enterprise Continuum is as a view of the repository of all the architecture assets. It can contain Architecture Descriptions, models, building blocks, patterns, architecture viewpoints, and other artifacts — that exist both within the enterprise and in the IT industry at large, which the enterprise considers to have available for the development of architectures for the enterprise.

Examples of internal architecture and solution artifacts are the deliverables of previous architecture work, which are available for re-use. Examples of external architecture and solution artifacts are the wide variety of industry reference models and architecture patterns that exist, and are continually emerging, including those that are highly generic (such as the TOGAF TRM); those specific to certain aspects of IT (such as a web services architecture, or a generic manageability architecture); those specific to certain types of information processing, such as e-Commerce, supply chain management, etc.; and those specific to certain vertical industries, such as the models generated by vertical consortia like the TM Forum (in the Telecommunications sector), ARTS (Retail), Energistics[®] (Petrotechnical), etc.

The Enterprise Architecture determines which architecture and solution artifacts an organization includes in its Architecture Repository. Re-use is a major consideration in this decision.

6.3 Constituents of the Enterprise Continuum

An overview of the context and constituents of the Enterprise Continuum is shown in Figure 6-1.

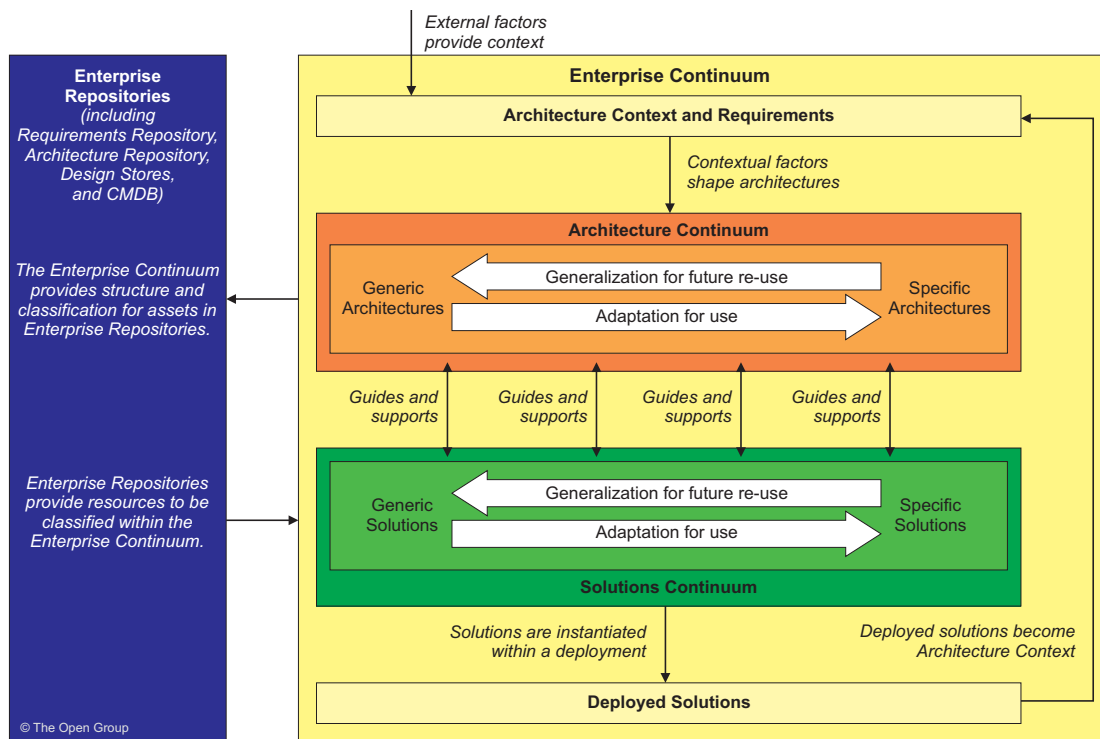


Figure 6-1 Enterprise Continuum

The Enterprise Continuum is partitioned into three distinct continua as follows:

- The **Enterprise Continuum** (see Section 6.4) is the outermost continuum and classifies assets related to the context of the overall Enterprise Architecture

The Enterprise Continuum classes of assets may influence architectures, but are not directly used during the ADM architecture development. The Enterprise Continuum classifies contextual assets used to develop architectures, such as policies, standards, strategic initiatives, organizational structures, and enterprise-level capabilities. The Enterprise Continuum can also classify solutions (as opposed to descriptions or specifications of solutions). Finally, the Enterprise Continuum contains two specializations, namely the Architecture and Solutions Continua.

- The **Architecture Continuum** (see Section 6.4.1) offers a consistent way to define and understand the generic rules, representations, and relationships in an architecture, including traceability and derivation relationships (e.g., to show that an Organization-Specific Architecture is based on an industry or generic standard)

The Architecture Continuum represents a structuring of Architecture Building Blocks (ABBs) which are re-usable architecture assets. ABBs evolve through their development lifecycle from abstract and generic entities to fully expressed Organization-Specific Architecture assets. The Architecture Continuum assets will be used to guide and select the elements in the Solutions Continuum (see below). The Architecture Continuum shows the relationships among foundational frameworks (such as the TOGAF framework), common system architectures (such as the III-RM), industry architectures, and Enterprise Architectures. The Architecture Continuum is a useful tool to discover commonality and eliminate unnecessary redundancy.

- The **Solutions Continuum** (see [Section 6.4.2](#)) provides a consistent way to describe and understand the implementation of the assets defined in the Architecture Continuum

The Solutions Continuum defines what is available in the organizational environment as re-usable SBBs. The solutions are the results of agreements between customers and business partners that implement the rules and relationships defined in the architecture space. The Solutions Continuum addresses the commonalities and differences among the products, systems, and services of implemented systems.

The Enterprise Continuum classifies architecture assets that are applicable across the entire scope of the Enterprise Architecture. These assets, which may be referred to as building blocks, can represent a variety of elements that collectively define and constrain the Enterprise Architecture. They can take the form of business goals and objectives, strategic initiatives, capabilities, policies, standards, and principles.

The Enterprise Continuum also contains the Architecture Continuum and the Solutions Continuum. Each of these continua is described in greater detail in the following sections.

6.4 Enterprise Continuum in Detail

The Enterprise Continuum is intended to represent the classification of all assets that are available to an enterprise. It classifies assets that exist within the enterprise along with other assets in the wider environment that are relevant to the enterprise, such as products, research, market factors, commercial factors, business strategies, and legislation.

The TOGAF Standard is intended to be a framework for conducting Enterprise Architecture and as a result many of the assets that reside within the Enterprise Continuum are beyond the specific consideration of the TOGAF framework. However, architectures are fundamentally shaped by concerns outside the practice of architecture and it is therefore of paramount importance that any architecture must accurately reflect external context.

The specific contextual factors to be identified and incorporated in an architecture will vary from architecture to architecture. However, typical contextual factors for architecture development are likely to include:

- External influencing factors, such as regulatory change, technological advances, and competitor activity
- Business strategy and context, including mergers, acquisitions, and other business transformation requirements
- Current business operations, reflecting deployed architectures and solutions

By observing the context for architecture, it can be seen that architecture development activity exists within a wider enterprise lifecycle of continuous change.

ABBs are defined in relation to a set of contextual factors and then realized through SBBs. SBBs are deployed as live solutions and become a part of the baseline operating model of the enterprise. The operating model of the enterprise and empiric information on the performance of the enterprise shapes the context and requirements for future change. Finally, these new requirements for change create a feedback loop to influence the creation of new Target Architectures.

6.4.1 Architecture Continuum

The Architecture Continuum illustrates how architectures are developed and evolved across a continuum ranging from Foundation Architectures, such as the TOGAF® Series Guide: The TOGAF® Technical Reference Model (TRM) through Common Systems Architectures, and Industry Architectures, and to an enterprise's own Organization-Specific Architectures.

The arrows in the Architecture Continuum represent the relationship that exists between the different architectures in the Architecture Continuum. The leftwards direction focuses on meeting enterprise needs and business requirements, while the rightwards direction focuses on leveraging architectural components and building blocks.

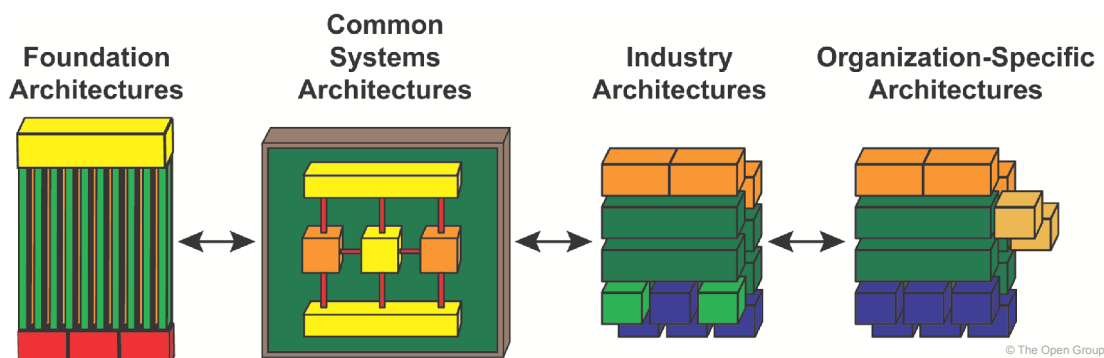


Figure 6-2 Architecture Continuum

The enterprise needs and business requirements are addressed in increasing detail from left to right. The architect will typically look to find re-usable architectural elements toward the left of the continuum. When elements are not found, the requirements for the missing elements are passed to the left of the continuum for incorporation. Those implementing architectures within their own organizations can use the same continuum models specialized for their business.

The four particular architecture types illustrated in [Figure 6-2](#) are intended to indicate the range of different types of architecture that may be developed at different points in the continuum; they are not fixed stages in a process.

Many different types of architecture may occur at points in between those illustrated in [Figure 6-2](#). Although the evolutionary transformation continuum illustrated does not represent a formal process, it does represent a progression, which occurs at several levels:

- Logical to physical

- Horizontal (IT-focused) to vertical (business-focused)
- Generalization to specialization
- Taxonomy to complete and specific architecture specification

At each point in the continuum, an architecture is designed in terms of the design concepts and building blocks available and relevant to that point.

The four architectures illustrated in [Figure 6-2](#) represent main classifications of potential architectures, and will be relevant and familiar to many architects. They are analyzed in detail below.

Foundation Architecture

A Foundation Architecture consists of generic components, inter-relationships, principles, and guidelines that provide a foundation on which more specific architectures can be built. The TOGAF ADM is a process that would support specialization of such Foundation Architectures in order to create organization-specific models.

The TOGAF TRM is an example of a Foundation Architecture. It is a fundamental architecture upon which other, more specific architectures can be based. See the TOGAF® Series Guide: The TOGAF® Technical Reference Model (TRM) for more details.

Common Systems Architectures

Common Systems Architectures guide the selection and integration of specific services from the Foundation Architecture to create an architecture useful for building common (i.e., highly re-usable) solutions across a wide number of relevant domains.

Examples of Common Systems Architectures include: a security architecture, a management architecture, a network architecture, an operations architecture, etc. Each is incomplete in terms of overall system functionality, but is complete in terms of a particular problem domain (security, manageability, networking, operations, etc.), so that solutions implementing the architecture constitute re-usable building blocks for the creation of functionally complete operating states of the enterprise.

Other characteristics of Common Systems Architectures include:

- Reflects requirements specific to a generic problem domain
- Defines building blocks specific to a generic problem domain
- Defines business, data, application, or technology standards for implementing these building blocks
- Provides building blocks for easy re-use and lower costs

The TOGAF Integrated Information Infrastructure Reference Model (III-RM) — see the TOGAF® Series Guide: The TOGAF Integrated Information Infrastructure Reference Model (III-RM) — is a reference model that supports describing Common Systems Architecture in the Application Domain that focuses on the requirements, building blocks, and standards relating to the vision of Boundaryless Information Flow.

Industry Architectures

Industry 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 capabilities and processes specific to a particular vertical industry, such as the Retail industry's "Active Store" architecture, or an Industry Architecture that incorporates the Energistics Data Model (refer to www.energistics.org).

Other characteristics of Industry Architectures include:

- Reflects requirements and standards specific to a vertical industry
- Defines building blocks specific to a generic problem domain
- Contains industry-specific logical data and process models
- Contains industry-specific applications and process models, as well as industry-specific business rules
- Provides guidelines for testing collections of systems
- Encourages levels of interoperability throughout the industry

Organization-Specific Architectures

Organization-Specific Architectures describe and guide the final deployment of solution components for a particular enterprise or extended network of connected enterprises.

There may be a variety of Organization-Specific Architectures that are needed to effectively cover the organization's requirements by defining the architectures in increasing levels of detail. Alternatively, this might result in several more detailed Organization-Specific Architectures for specific entities within the global enterprise. Breaking down Organization-Specific Architectures into constituent pieces is addressed in the TOGAF Standard — Applying the ADM.

The Organization-Specific Architecture guides the final customization of the solution, and has the following characteristics:

- Provides a means to communicate and manage business operations across all four architecture domains
- Reflects requirements specific to a particular enterprise
- Defines building blocks specific to a particular enterprise
- Contains organization-specific business models, data, applications, and technologies
- Provides a means to encourage implementation of appropriate solutions to meet business needs
- Provides the criteria to measure and select appropriate products, solutions, and services
- Provides an evolutionary path to support growth and new business needs

6.4.2 Solutions Continuum

The Solutions Continuum represents the detailed specification and construction of the architectures at the corresponding levels of the Architecture Continuum. At each level, the Solutions Continuum is a population of the architecture with reference building blocks — either purchased products or built components — that represent a solution to the enterprise's business need expressed at that level. A populated repository based on the Solutions Continuum can be regarded as a solutions inventory or re-use library, which can add significant value to the task of managing and implementing improvements to the enterprise.

The Solutions Continuum is illustrated in [Figure 6-3](#).

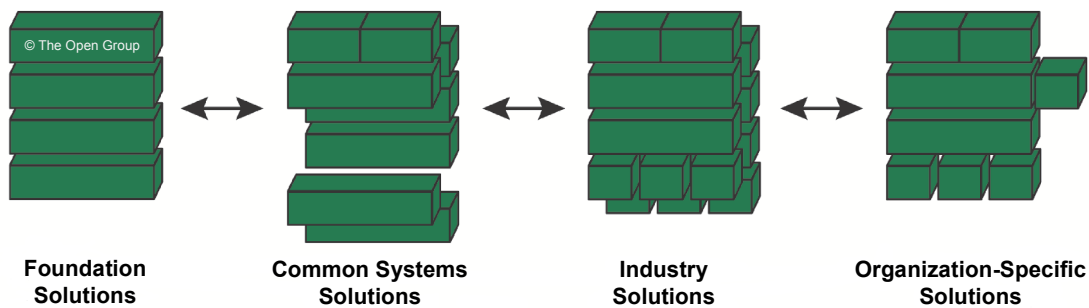


Figure 6-3 Solutions Continuum

"Moving to the right" on the Solutions Continuum is focused on providing solutions value (i.e., foundation solutions provide value in creating common systems solutions; common systems solutions are used to create industry solutions; and industry solutions are used to create organization-specific solutions). "Moving to the left" on the Solutions Continuum is focused on addressing enterprise needs. These two viewpoints are significant for a company attempting to focus on its needs while maximizing the use of available resources through leverage.

The following subsections describe each of the solution types within the Solutions Continuum.

Foundation Solutions

Foundation Solutions are highly generic concepts, tools, products, services, and solution components that are the fundamental providers of capabilities. Services include professional services — such as training and consulting services — that ensure the maximum investment value from solutions in the shortest possible time; and support services — such as Help Desk — that ensure the maximum possible value from solutions (services that ensure timely updates and upgrades to the products and systems).

Example Foundation Solutions would include programming languages, operating systems, foundational data structures (such as EDIFACT), generic approaches to organization structuring, foundational structures for organizing IT operations (such as ITIL® or the IT4IT Reference Architecture), etc.

Common Systems Solutions

A Common Systems Solution is an implementation of a Common Systems Architecture comprised of a set of products and services, which may be certified or branded. It represents the highest common denominator for one or more solutions in the industry segments that the Common Systems Solution supports.

Common Systems Solutions represent collections of common requirements and capabilities, rather than those specific to a particular customer or industry. Common Systems Solutions provide organizations with operating environments specific to operational and informational needs, such as high availability transaction processing and scalable data warehousing systems. Examples of Common Systems Solutions include an enterprise management system product or a security system product.

Computer systems vendors are the typical providers of technology-centric Common Systems Solutions. "Software as a service" vendors are typical providers of common application solutions. Business process outsourcing vendors are typical providers of business capability-centric Common Systems Solutions.

Industry Solutions

An Industry Solution is an implementation of an Industry Architecture, which provides re-usable packages of common components and services specific to an industry.

Fundamental components are provided by Common Systems Solutions and/or Foundation Solutions, and are augmented with industry-specific components. Examples include: a physical database schema or an industry-specific point-of-service device.

Industry Solutions are industry-specific, aggregate procurements that are ready to be tailored to an individual organization's requirements.

In some cases an industry solution may include not only an implementation of the Industry Architecture, but also other solution elements, such as specific products, services, and systems solutions that are appropriate to that industry.

Organization-Specific Solutions

An Organization-Specific Solution is an implementation of the Organization-Specific Architecture that provides the required business capabilities. Because solutions are designed for specific business operations, they contain the highest amount of unique content in order to accommodate the varying people and processes of specific organizations.

Building Organization-Specific Solutions on Industry Solutions, Common Systems Solutions, and Foundation Solutions is the primary purpose of connecting the Architecture Continuum to the Solutions Continuum, as guided by the architects within an enterprise.

An Organization-Specific Solution will be structured in order to support specific SLAs to ensure support of the operational systems at desired service levels. For example, a third-party application hosting provider may offer different levels of support for operational systems. These agreements would define the terms and conditions of that support.

Other key factors to be defined within an Organization-Specific Solution are the key operating parameters and quality metrics that can be used to monitor and manage the environment.

The Enterprise Continuum can provide a key link between architecture, development, and operations personnel by allowing them to communicate and reach agreement on anticipated operational support requirements. Operations personnel can in turn access the Enterprise

Continuum to obtain information regarding the operation concepts and service support requirements of the deployed system.

6.5 The Enterprise Continuum and the ADM

The TOGAF ADM describes the process of developing an enterprise-specific architecture and an enterprise-specific solution(s) which conform to that architecture by adopting and adapting (where appropriate) generic architectures and solutions (left to right in the continuum classification). In a similar fashion, specific architectures and solutions that prove to be credible and effective will be generalized for re-use (right to left in the continuum classification).

At relevant places throughout the TOGAF ADM, there are pointers to useful architecture assets at the relevant level of generality in the continuum classification. These assets can include reference models from The Open Group and industries at large.

The TOGAF Library provides reference models for consideration for use in developing an organization's architecture.

However, in developing architectures in the various domains within an overall Enterprise Architecture, the architect will need to consider the use and re-use of a wide variety of different architecture assets, and the Enterprise Continuum provides an approach for categorizing and communicating these different assets.

6.6 The Enterprise Continuum and Your Organization

The preceding sections have described the Enterprise Continuum, the Architecture Continuum, and the Solutions Continuum. The following sections describe the relationships between each of the three continua and how these relationships should be applied within your organization.

6.6.1 Relationships

Each of the three continua contains information about the evolution of the architectures during their lifecycle:

- The Enterprise Continuum provides an overall context for architectures and solutions, and classifies assets that apply across the entire scope of the enterprise
- The Architecture Continuum provides a classification mechanism for assets that collectively define the architecture at different levels of evolution from generic to specific
- The Solutions Continuum provides the classification for assets to describe specific solutions for the organization that can be implemented to achieve the intent of the architecture

The relationships between the Architecture Continuum and Solutions Continuum are shown in [Figure 6-4](#).

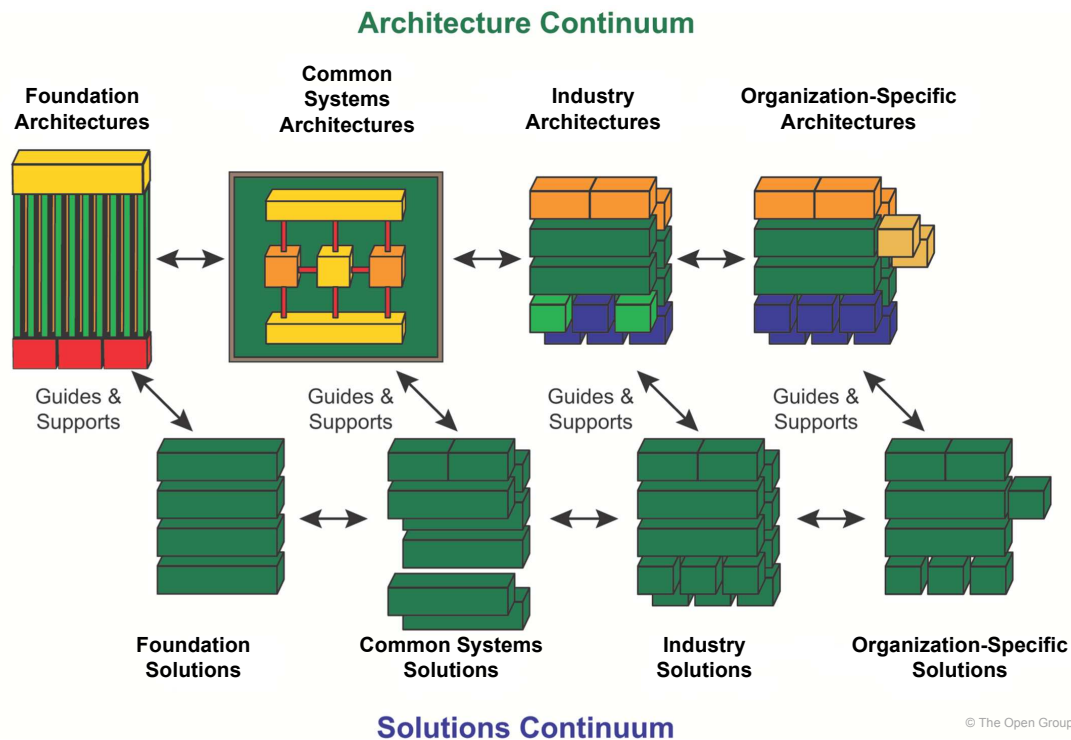


Figure 6-4 Relationships between Architecture and Solutions Continua

The relationship between the Architecture Continuum and the Solutions Continuum is one of guidance, direction, and support. For example, Foundation Architectures guide the creation or selection of Foundation Solutions. Foundation Solutions support the Foundation Architecture by helping to realize the architecture defined in the Architecture Continuum. The Foundation Architecture also guides development of Foundation Solutions, by providing architectural direction, requirements, and principles that guide selection, and realization of appropriate solutions. A similar relationship exists between the other elements of the Enterprise Continuum.

The Enterprise Continuum presents mechanisms to help improve productivity through leverage. The Architecture Continuum offers a consistent way to understand the different architectures and their components. The Solutions Continuum offers a consistent way to understand the different products, systems, services, and solutions required.

The Enterprise Continuum should not be interpreted as representing strictly chained relationships. Organization-Specific Architectures could have components from a Common Systems Architecture, and Organization-Specific Solutions could contain Foundation Solutions. The relationships depicted in Figure 6-1 are an illustration showing opportunities for leveraging architecture and solution components.

6.6.2 Your Enterprise

The TOGAF Standard provides a method for you to "architect" the systems in your enterprise. Your architecture organization will have to deal with each type of architecture described above. For example, it is recommended that you have your own Foundation Architecture that governs all of your systems. You should also have your own Common Systems Architectures that govern major shared systems — such as the networking system or management system. You may have your own industry-specific architectures that govern the way your systems must behave within your industry. Finally, any given department or organization within your business may need its own individual Organization-Specific Architecture to govern the systems within that department.

Your architecture organization will either adopt or adapt existing architectures, or will develop its own architectures from the ground up. In either case, the TOGAF Standard is a tool to help. It provides a method to assist you in generating/maintaining any type of architecture within the Architecture Continuum while leveraging architecture assets already defined, internal or external to your organization. The TOGAF ADM helps you to re-use architecture assets, making your architecture organization more efficient and effective.

Chapter 7: Architecture Repository

7.1 Overview

Operating a mature Architecture Capability within a large enterprise creates a huge volume of architectural output. Effective management and leverage of these architectural work products require a formal taxonomy for different types of architectural asset alongside dedicated processes and tools for architectural content storage.

This section provides a structural framework for an Architecture Repository that allows an enterprise to distinguish between different types of architectural assets that exist at different levels of abstraction in the organization. This Architecture Repository is one part of the wider Enterprise Repository, which provides the capability to link architectural assets to components of the Detailed Design, Deployment, and Service Management Repositories.

At a high level, the following 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 Library** 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 Repository** provides a record of governance activity across the enterprise
- The **Architecture Requirements Repository** provides a view of all authorized architecture requirements which have been agreed with the Architecture Board
- The **Solutions Landscape** presents an architectural representation of the SBBs supporting the Architecture Landscape which have been planned or deployed by the enterprise

The relationships between these areas of the Architecture Repository are shown in [Figure 7-1](#).

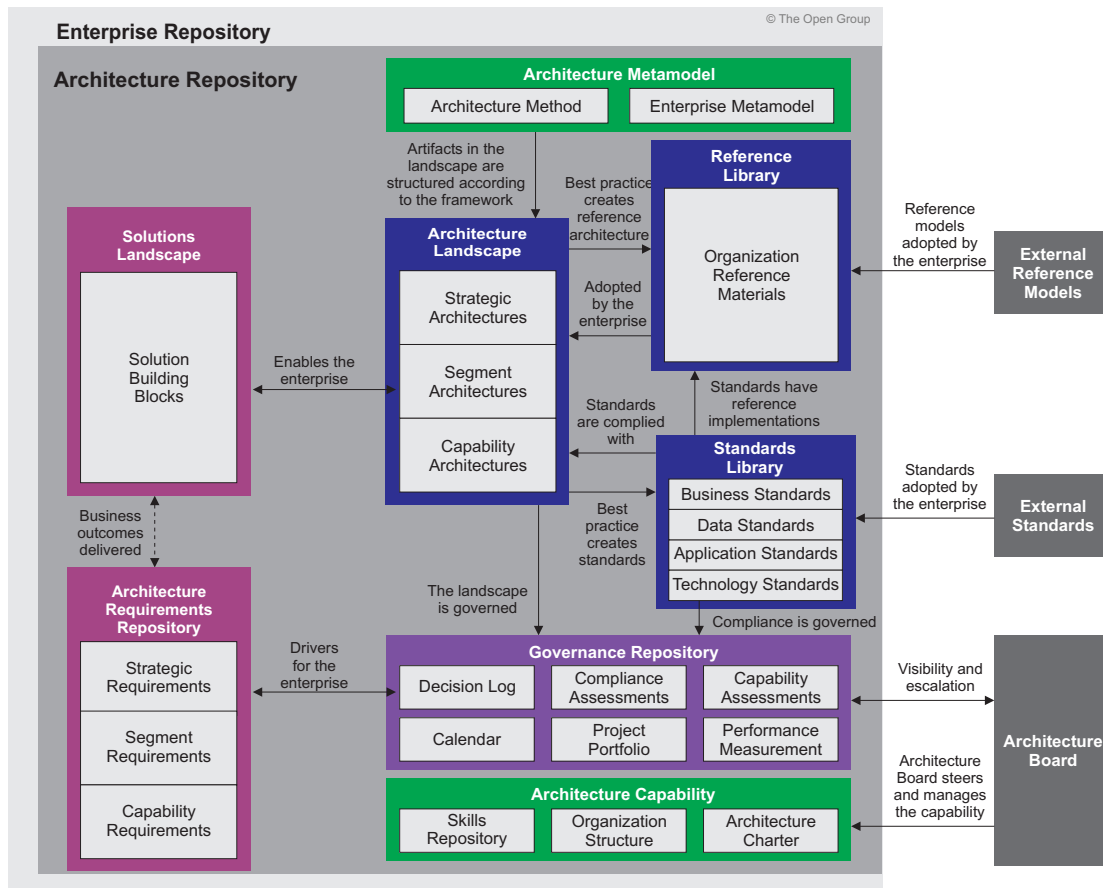


Figure 7-1 Overview of Architecture Repository

The following sections describe the structure and content of the repository areas.

7.2 Architecture Landscape

The 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** (see the TOGAF Standard — Architecture Development Method) 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** (see the TOGAF Standard — Architecture Development Method) 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** (see the TOGAF Standard — Architecture Development Method) 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.

7.3 Reference Library

7.3.1 Overview

The Reference Library provides a repository to hold reference materials that should be used to develop architectures. Reference materials held may be obtained from a variety of sources, including:

- Standards bodies
- Product and service vendors
- Industry communities or forums
- Standard templates
- Enterprise best practice

The Reference Library should contain:

- Reference Architectures
- Reference Models
- Viewpoint Library
- Templates

Note: 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. Typically, a generic reference architecture provides the architecture team with an outline of their organization-specific reference architecture that will be customized for a specific organization. For example, a generic reference architecture may identify that there is a need for data models. An example of a reference architecture is the IT4IT Reference Architecture which also defines a common information model for IT management. Another example is the TM Forum eTOM and SID as an organization-specific reference architecture.

In order to segregate different classes of architecture reference materials, the Reference Library can use the Architecture Continuum as a method for classification.

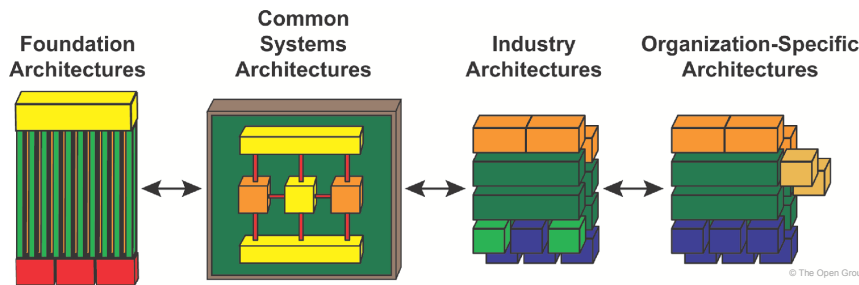


Figure 7-2 Architecture Continuum

The Architecture Continuum, as shown in [Figure 7-2](#), can be viewed as a Reference Library classification scheme. As such it illustrates how reference architectures can be organized across a range — from Foundation Architectures, and Industry-Specific Architectures, to an Organization-Specific Architecture.

The enterprise needs and business requirements are addressed in decreasing abstraction from left to right. The architect will typically find more re-usable architectural elements toward the left of the range. When elements are not found, the requirements for the missing elements are passed to the left of the range for incorporation.

Through this exercise it is important to keep in mind the concepts of levels and partitions. At different levels of granularity there may exist reference materials appropriate to the level, and partitions within the Architecture Landscape can be expected to use different reference material (see the TOGAF Standard — Applying the ADM).

7.4 Standards Library

7.4.1 Overview

The Standards Library provides a repository area to hold a set of specifications, to which architectures must conform. Establishment of a Standards Library provides an unambiguous basis for Architecture Governance because:

- The standards are easily accessible to projects and therefore the obligations of the project can be understood and planned for
- Standards are stated in a clear and unambiguous manner, so that compliance can be objectively assessed

7.4.2 Types of Standard

Standards typically fall into three classes:

- **Legal and Regulatory Obligations:** these standards are mandated by law and therefore an enterprise must comply or face serious consequences
- **Industry Standards:** these standards are established by industry bodies, such as The Open Group, and are then selected by the enterprise for adoption

Industry Standards offer potential for interoperation and sharing across enterprises, but also fall outside of the control of the enterprise and therefore must be actively monitored.
- **Organizational Standards:** these standards are set within the organization and are based on business aspiration (e.g., selection of standard applications to support portfolio consolidation)

Organizational Standards require processes to allow for exemptions and standards evolution.

7.4.3 Standards Lifecycle

Standards do not generally exist for all time. New standards are identified and managed through a lifecycle process.

Typically, standards pass through the following stages:

- **Proposed Standard:** a potential standard has been identified for the organization, but has not yet been evaluated for adoption
- **Provisional Standard** (also known as a **Trial Standard**): a Provisional Standard has been identified as a potential standard for the organization, but has not been tried and tested to a level where its value is fully understood

Projects wishing to adopt Provisional Standards may do so, but under specific pilot conditions, so that the viability of the standard can be examined in more detail.
- **Standard** (also known as an **Active Standard**): a Standard defines a mainstream solution that should generally be used as the approach of choice
- **Phasing-Out Standard** (also known as a **Deprecated Standard**): a Phasing-Out Standard is approaching the end of its useful lifecycle

Projects that are re-using existing components can generally continue to make use of Phasing-Out Standards. Deployment of new instances of the Phasing-Out Standard is generally discouraged.
- **Retired Standard** (also known as an **Obsolete Standard**): a Retired Standard is no longer accepted as valid within the landscape

In most cases, remedial action should be taken to remove the Retired Standard from the landscape. Change activity on a Retired Standard should only be accepted as a part of an overall decommissioning plan.

All standards should be periodically reviewed to ensure that they sit within the right stage of the standards lifecycle. As a part of standards lifecycle management, the impact of changing the lifecycle status should be addressed to understand the landscape impact of a standards change and plan for appropriate action to address it.

7.4.4 Standards Classification within the Standards Library

Standards within the Standards Library are categorized according to the building blocks within the TOGAF Enterprise Metamodel. Each metamodel entity can potentially have standards associated with it (e.g., Business Service, Technology Component).

Standards may relate to "approved" building blocks (e.g., a list of standard technology components) or may specify appropriate use of a building block (e.g., scenarios where messaging infrastructure is appropriate, application communication standards are defined).

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

- **Business Standards:**

- Standard shared business capabilities
- 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

7.5 Governance Repository

7.5.1 Overview

The Governance Repository provides a repository area to hold shared information relating to the ongoing governance of projects. Maintaining a shared repository of governance information is important, because:

- Decisions made during projects (such as standards deviations or the rationale for a particular architectural approach) are important to retain and access on an ongoing basis

For example, if a system is to be replaced, having sight of the key architectural decisions that shaped the initial implementation is highly valuable as it will highlight constraints that may otherwise be obscured.

- Many stakeholders are interested in the outcome of project governance (e.g., other projects, customers of the project, the Architecture Board, etc.)

7.5.2 Contents of the Governance Repository

The Governance Repository should contain the following items:

- **Decision Log:** a log of all architecturally significant decisions that have been made in the organization

This would typically include:

- Product selections
- Justification for major architectural features of projects
- Standards deviations
- Standards lifecycle changes
- Change Request evaluations and approvals
- Re-use assessments

- **Compliance Assessments:** at key checkpoint milestones in the progress of a project, a formal architecture review will be carried out

This review will measure the compliance of the project to the defined architecture standards. For each project, this log should include:

- Project overview
- Progress overview (timeline, status, issues, risks, dependencies, etc.)
- Completed architecture checklists
- Standards compliance assessment
- Recommended actions

- **Capability Assessments:** depending on their objectives, some projects will carry out assessments of business, IT, or Architecture Capability

These assessments should be periodically carried out and tracked to ensure that appropriate progress is being made. This log should include:

- Templates and reference models for executing Capability Assessments
- Business Capability Assessments
- IT capability, maturity, and impact assessments
- Architecture maturity assessments
- **Calendar:** the Calendar should show a schedule of in-flight projects and formal review sessions to be held against these projects
- **Project Portfolio:** the Project Portfolio should hold summary information about all in-flight projects that fall under Architecture Governance, including:
 - The name and description of the project
 - Architectural scope of the project
 - Architectural roles and responsibilities associated with the project
- **Performance Measurement:** based on a charter for the architecture function, a number of performance criteria will typically be defined

The Performance Measurement log should capture metrics relating to project governance and any other performance metrics relating to the architecture charter so that performance can be measured and evaluated on an ongoing basis.

7.6 The Architecture Requirements Repository

7.6.1 Overview

The Architecture Requirements Repository is used by all phases of the ADM to record and manage all information relevant to the architecture requirements. The requirements address the many types of architecture requirements; i.e., strategic, segment, and capability requirements which are the major drivers for the Enterprise Architecture.

Requirements can be gathered at every stage of the architecture development cycle and need to be approved through the various phases and governance processes.

The Requirements Management phase is responsible for the management of the contents of the Architecture Requirements Repository and ensuring the integrity of all requirements and their availability for access by all phases.

7.6.2 Contents of the Architecture Requirements Repository

The Architecture Requirements Repository holds architectural requirements of the required state of the enterprise at particular points in time. Due to the sheer volume and the diverse stakeholder needs throughout the architecture development cycle, the Architecture Requirements are divided into three levels of granularity:

1. **Strategic Architecture Requirements** show a long-term summary view of the requirements for the entire enterprise.

Strategic Architecture Requirements identify operational and change requirements for direction setting at an executive level.

2. **Segment Architecture Requirements** provide more detailed operating model requirements for areas within an enterprise.

Segment Architecture Requirements may identify requirements at the program or portfolio level to identify and align more detailed change activity.

3. **Capability Architecture Requirements** identify the detailed requirements for a particular unit of capability.

Capability Architecture Requirements identify requirements for individual work packages and projects to be grouped within managed portfolios and programs.

The business outcomes for architecture requirements will be reflected in the Solutions Landscape over time. When this occurs, the architecture requirements are met and archived for audit purposes.

7.7 Solutions Landscape

The Solutions Landscape holds the SBBs which support the ABBS specified, developed, and deployed. The building blocks may be products or services which may be categorized according to the Enterprise Continuum categorization and/or the ABB specifications as Strategic, Segment, or Capability SBBs.

SBBs may also include tools, systems, services, and information which describe the actual solutions that may be selected and their operation. For example, vendor-specific reference models or vendor-specific Levels 4 and 5 of the IT4IT Reference Architecture would be defined here.

However, the Solutions Landscape will not include the information and data content produced by the solutions selected; that is the responsibility of the solutions themselves.

7.8 The Enterprise Repository

While the Architecture Repository holds information concerning the Enterprise Architecture and associated specifications and artifacts, there are a considerable number of enterprise repositories that support the architecture both inside and outside of the enterprise.

These can include development repositories, specific operating environments, instructions, and configuration management repositories.

7.9 External Repositories

7.9.1 External Reference Models

There are many industry reference models available which may assist in understanding the role of and developing the reference architectures.

7.9.2 External Standards

These relate to industry, best practice, or formal defined standards used by leading organizations. Examples include ISO, IEEE, and Government standards.

7.9.3 Architecture Board Approvals

Decisions made by the Architecture Board which affect the Enterprise Architecture are often recorded in the minutes of meetings. These minutes are often held in documentation archives which are excluded from the Architecture Repository for legal or regulatory reasons.

Index

ABB	65, 79-80
Active Store.....	90
Actor/Role matrix	46
ADM.....	36, 82
Application and User Location diagram	55
Application Communication diagram	55
Application Interaction matrix.....	55
Application Migration diagram.....	57
Application Portfolio catalog.....	52
Application Use-Case diagram	56
Application/Data matrix.....	50
Application/Function matrix.....	54
Application/Organization matrix	53
Application/Technology matrix	58
Applications Standards	102
architectural artifact	31
concepts.....	31
architecture artifacts	39
architecture building block	80
Architecture Capability	97
Architecture Continuum	80, 88
Architecture Contract	65
Architecture Definition Document	66
architecture deliverables	63
Architecture Description.....	31
Architecture Landscape	97-98
Architecture Metamodel	97
architecture model	31
Architecture Principles	67
Architecture Repository	8, 68, 97
Architecture Requirements Repository	97, 104
Architecture Requirements Specification	68
Architecture Roadmap	69
architecture view	31
creation	35
examples	36
guidelines	34
architecture viewpoint	32
examples	36
Architecture Vision	70
Benefits diagram	61
building blocks	39, 79
characteristics	79
design.....	82

Business Capabilities catalog	45
Business Capability map	43, 49
business drivers	70
Business Event diagram	49
Business Footprint diagram	47
Business Glossary catalog	46
business goals	70
Business Interaction matrix.....	46
Business Model diagram.....	42
business principles	70
business scenarios	70
Business Service/Function catalog	44
Business Service/Information diagram	47
Business Standards	102
Business Use-Case diagram	48
Calendar	104
Capability Architecture	99
Capability Architecture Requirements	105
Capability Assessment	71, 103
Capability/Organization matrix	47
catalog	39
catalogs	39
Change Request	72
CMMI	63
Common Systems Architectures	89
Common Systems Solutions.....	92
Communications Plan	73
Compliance Assessment	73, 103
Conceptual Data diagram	51
concern	33
concerns	31
Content Framework	1, 3-5, 9
Content Framework, and the TOGAF ADM	7
Contract/Measure catalog.....	45
COTS	82
CRM.....	50
CRUD.....	39
Data Dissemination diagram.....	51
Data Entity/Business Function matrix	50
Data Entity/Data Component catalog	49
Data Lifecycle diagram	52
Data Migration diagram.....	52
Data Security diagram	51
Data Standards.....	102
Decision Log	103
deliverables	64
Deployment Repository	97
Detailed Design Repository	97
diagrams	39
Driver/Goal/Objective catalog	43

Energistics	90
Enterprise Continuum	8, 85
Enterprise Manageability diagram	56
Enterprise Metamodel	3-4, 9
attributes	15
detail	11
entities	12
overview	10
relationships	25
vision	9
Enterprise Repository	97, 105
Environment diagram	58
eTOM	99
Foundation Architecture	89
Foundation Solutions	91
Functional Decomposition diagram	47
Goal/Objective/Business Service diagram	48
Governance Repository	97, 103
contents of	103
III-RM	89
Implementation and Migration Plan	74
Implementation Governance Model	75
Industry Architectures	90
Industry Solutions	92
information map	49
Interface catalog	52
IT4IT Reference Architecture	91, 99, 105
ITIL	91
Location catalog	44
Location diagram	58
Logical Data diagram	51
matrices	39
matrix	39
model kind	32
MSP	63
Network and Communications diagram	61
Networked Computing/Hardware diagram	60
Organization Decomposition diagram	48
organization map	49
Organization-Specific Architectures	90
Organization-Specific Solutions	92
Organization/Actor catalog	43
Organizational Model	75
performance measurement	104
Platform Decomposition diagram	59
PMBOK	63
PRINCE2	63
Principles catalog	41
Process Flow diagram	48
Process/Application Realization diagram	56

Process/Event/Control/Product catalog	45
Processing diagram	59
Product Lifecycle diagram.....	48
Project Context diagram	61
Project Portfolio.....	104
Reference Library	97, 99
Request for Architecture Work.....	76
Requirements catalog	61
Requirements Impact Assessment	76
Role catalog	44
Role/Application matrix	54
SBB.....	77, 79, 81
Segment Architecture	98
Segment Architecture Requirements.....	105
Service Management Repository	97
SID	99
SMART	12
Software Distribution diagram.....	57
Software Engineering diagram.....	57
solution building blocks	81
Solution Concept diagram.....	42
Solutions Continuum.....	81, 91
Solutions Landscape	97, 105
stakeholder	31
Stakeholder catalog	42
Standards Library	97, 100, 102
standards, classification.....	102
standards, lifecycle	101
standards, types of.....	101
Statement of Architecture Work.....	77
Strategic Architecture.....	98
Strategic Architecture Requirements	104
Strategy/Capability matrix.....	47
Tailored Architecture Framework	78
Technology Portfolio catalog	58
Technology Standards	102
Technology Standards catalog.....	57
TM Forum	99
TOGAF ADM.....	63, 93
TOGAF TRM.....	89
Value Chain diagram.....	42
Value Stream catalog.....	45
value stream map	43, 49
Value Stream Stages catalog.....	46
Value Stream/Capability matrix.....	47

The Open Group Standard

TOGAF® Standard — Enterprise Architecture Capability and Governance

The Open Group

Copyright © 2005-2022, The Open Group

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

Any use of this publication for commercial purposes is subject to the terms of the Annual Commercial License relating to it. For further information, see www.opengroup.org/legal/licensing.

The Open Group Standard

TOGAF® Standard — Enterprise Architecture Capability and Governance

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

Chapter	1	Introduction	1
Chapter	2	Establishing an Architecture Capability	3
	2.1	Overview.....	3
	2.2	Phase A: Architecture Vision	4
	2.3	Phase B: Business Architecture	5
	2.4	Phase C: Data Architecture	6
	2.5	Phase C: Application Architecture	6
	2.6	Phase D: Technology Architecture	6
	2.7	Phase E: Opportunities & Solutions	6
	2.8	Phase F: Migration Planning	6
	2.9	Phase G: Implementation Governance	6
	2.10	Phase H: Architecture Change Management	7
	2.11	Requirements Management	7
Chapter	3	Architecture Governance	9
	3.1	Introduction.....	9
	3.1.1	Levels of Governance within the Enterprise	9
	3.1.2	Nature of Governance	10
	3.1.3	Technology Governance	11
	3.1.4	IT Governance	11
	3.1.5	Architecture Governance: Overview	12
	3.2	Architecture Governance Framework	13
	3.2.1	Architecture Governance Framework — Conceptual Structure	13
	3.2.2	Architecture Governance Framework — Organizational Structure.....	16
	3.3	Architecture Governance in Practice	18
	3.3.1	Architecture Governance — Key Success Factors	18
	3.3.2	Elements of an Effective Architecture Governance Strategy	19
Chapter	4	Architecture Board.....	21
	4.1	Role	21
	4.2	Responsibilities.....	21
	4.3	Setting Up the Architecture Board	22
	4.3.1	Triggers.....	22
	4.3.2	Size of the Board	23
	4.3.3	Board Structure	24
	4.4	Operation of the Architecture Board	24
	4.4.1	General.....	24
	4.4.2	Preparation	25

	4.4.3	Agenda	25
Chapter	5	Architecture Contracts.....	27
	5.1	Role	27
	5.2	Contents	29
	5.2.1	Statement of Architecture Work.....	29
	5.2.2	Contract between Architecture Design and Development Partners	29
	5.2.3	Contract between Architecting Function and Business Stakeholders	30
	5.3	Relationship to Architecture Governance	30
Chapter	6	Architecture Compliance.....	31
	6.1	Introduction.....	31
	6.2	Terminology: The Meaning of Architecture Compliance	31
	6.3	Architecture Compliance Reviews	33
	6.3.1	Purpose	33
	6.3.2	Timing	34
	6.3.3	Governance and Personnel Scenarios	35
	6.4	Architecture Compliance Review Process	36
	6.4.1	Overview.....	36
	6.4.2	Roles	37
	6.4.3	Steps	38
	6.5	Architecture Compliance Review Checklists	39
	6.5.1	Hardware and Operating System Checklist.....	39
	6.5.2	Software Services and Middleware Checklist.....	40
	6.5.3	Applications Checklists	42
	6.5.4	Information Management Checklists	44
	6.5.5	Security Checklist	45
	6.5.6	System Management Checklist	46
	6.5.7	System Engineering/Overall Architecture Checklists	47
	6.5.8	System Engineering/Methods & Tools Checklist	50
	6.6	Architecture Compliance Review Guidelines	51
	6.6.1	Tailoring the Checklists	51
	6.6.2	Conducting Architecture Compliance Reviews.....	52
		Index	53

List of Figures

3-1	Architecture Governance Framework — Conceptual Structure	13
3-2	Architecture Governance Framework — Organizational Structure	16
6-1	Levels of Architecture Conformance.....	31
6-2	Architecture Compliance Review Process.....	36

Contents

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.

Preface

This Document

This is the TOGAF Standard — Enterprise Architecture Capability and Governance.

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.

COBIT is a registered trademark of the Information Systems Audit and Control Association (ISACA) and the IT Governance Institute.

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.

Referenced Documents

Chapter 1: Introduction

This chapter provides an introduction to the guidance provided in the TOGAF Standard — Enterprise Architecture Capability and Governance (this document).

In order to successfully operate an architecture function within an enterprise, it is necessary to put in place appropriate organization structures, processes, roles, responsibilities, and governance.

Guidelines included within this document are as follows:

- Establishing an Architecture Capability (see [Chapter 2](#)) provides guidelines on how to use the ADM to establish an Architecture Capability
- Architecture Governance (see [Chapter 3](#)) provides a framework and guidelines for Architecture Governance
- Architecture Board (see [Chapter 4](#)) provides guidelines for establishing and operating an Enterprise Architecture Board
- Architecture Contracts (see [Chapter 5](#)) provides guidelines for defining and using Architecture Contracts
- Architecture Compliance (see [Chapter 6](#)) provides guidelines for ensuring project compliance to the architecture

Chapter 2: Establishing an Architecture Capability

This chapter provides guidelines on how to use the ADM to establish an Architecture Capability.

2.1 Overview

As with any business capability, the establishment of an Enterprise Architecture Capability can be supported by the TOGAF Architecture Development Method (ADM). Successful use of the ADM will provide a customer-focused, value-adding, and sustainable architecture practice that enables the business, helps maximize the value of investments, and pro-actively identifies opportunities to gain business benefits and manage risk.

Establishing a sustainable architecture practice within an organization can be achieved by adhering to the same approach that is used to establish any other capability — such as a Business Process Management (BPM) capability — within an organization. The ADM is an ideal method to be used to architect and govern the implementation of such a capability. Applying the ADM with the specific Architecture Vision to establish an architecture practice within the organization would achieve this objective.

The establishment of the architecture practice should not be seen as a phase of an architecture project, or a one-off project, but rather as an ongoing discipline that provides the context, environment, and resources to govern and enable architecture delivery to the organization. As an architecture project is executed within this environment it might request a change to the architecture practice that would trigger another cycle of the ADM to extend the architecture practice.

Implementing any capability within an organization would require the design of the four domain architectures: Business, Data, Application, and Technology. Establishing the architecture practice within an organization would therefore require the design of:

- The **Business Architecture** of the architecture practice that will highlight the Architecture Governance, architecture processes, architecture organizational structure, architecture information requirements, architecture products, etc.
- The **Data Architecture** that would define the structure of the organization's Enterprise Continuum and Architecture Repository
- The **Application Architecture** specifying the functionality and/or applications services required to enable the architecture practice
- The **Technology Architecture** that depicts the architecture practice's infrastructure requirements and deployment in support of the architecture applications and Enterprise Continuum

The steps in establishing an architecture practice are explained below, against the context of the ADM phases. The reader should therefore refer to the relevant ADM phase in the TOGAF Standard — Architecture Development Method to understand the complete scope of each step.

In this section, key aspects will be highlighted for each ADM phase that should be considered and are specific to establishing an architecture practice. The intent is therefore not to repeat each ADM phase description, but to guide the reader to apply each ADM phase within the context of establishing an architecture practice.

2.2 Phase A: Architecture Vision

The purpose of this phase within the context of establishing an architecture practice is to define or review the vision, stakeholders, and principles of the architecture practice. The focus in this phase would be on the architecture practice as a whole and not on a particular architecture project.

The following should be considered in terms of understanding the steps in the context of establishing an architecture practice:

- **Establish the Project:** this step should focus on defining the stakeholders in the architecture practice

The stakeholders would include the roles and organization units participating in the architecture practice, as well as people who will benefit from the deliverables generated by the architecture practice and who can therefore be defined as customers of the architecture practice.

- **Identify Stakeholders and Concerns, Business Requirements, and Architecture Vision:** this step generates the first, very high-level definitions of the baseline and target environments, from the perspectives of business information systems and technology, for the architecture practice
- **Identify Business Goals and Business Drivers:** an understanding of the business goals and drivers is essential to align the architecture practice to the business
- **Define Scope:** defining the scope of the architecture practice is a high-level project plan of what is scheduled to be addressed in terms of architecture for the next period
- **Define Constraints:** the focus in this step is the enterprise-wide constraints that impact on all architecture projects
- **Review Architecture Principles, including Business Principles:** the intent in this step is to define the principles that govern and guide the running of the architecture practice

Where Architecture Principles usually govern the architecture deliverables, the architecture practice principles address the architecture practice organization, content, tools, and process.
- **Develop Statement of Architecture Work and Secure Approval:** this step generates the architecture practice vision and scope

Another step that can be considered during this phase is to conduct an architecture maturity assessment. Refer to the TOGAF® Series Guide: Architecture Maturity Models for guidance on this topic.

2.3 Phase B: Business Architecture

Key areas of focus during this phase of establishing or refining the Business Architecture of the architecture practice are:

- An **Architecture Ontology** defining the architectural terms and definitions that will be used in the organization in order to establish a common understanding of these terms
- The **Architecture Process** where the ADM would form the base of the process and need to be customized to meet the organization's requirements and architecture practice vision

Refer to the TOGAF Standard — Architecture Development Method for guidance on developing this process. The required Architecture Governance processes should be included in the overall architecture process.

- The **Architecture Viewpoints and Views** that list all the viewpoints and views that should be addressed by the architecture practice

The identified architecture practice stakeholders would guide the development of this definition. One of the viewpoints to be included is the Architecture Governance viewpoint; refer to the TOGAF Standard — Architecture Content for guidance on this output.

- The **Architecture Framework** describing the various architecture deliverables that will be generated by the architecture practice, the inter-relationships and dependencies between the architecture deliverables, as well as the rules and guidelines governing the design of these deliverables

The defined architecture viewpoints and views should be used to guide the definition of the architecture framework. The TOGAF Standard — Architecture Development Method and the TOGAF Standard — Architecture Content are useful references that will assist in describing the architecture framework.

- The **Architecture Accountability Matrix** defining the roles in the architecture practice and allocating accountability of the roles to architecture deliverables and processes

This matrix would include the required Architecture Governance structures and roles. The TOGAF Standard — Architecture Development Method as well as [Chapter 4](#), [Chapter 3](#), and the TOGAF® Series Guide: Architecture Skills Framework would provide guidance on this output.

- The **Architecture Performance Metrics** identifying and describing the metrics that will be used to monitor the performance of the architecture practice against its stated architecture practice vision and objectives
- The **Architecture Governance Framework** which is a specific view of the defined architecture process and Architecture Accountability Matrix

2.4 Phase C: Data Architecture

The Data Architecture of the architecture practice would specify and govern the structure of the organization's Enterprise Continuum and Architecture Repository. The Data Architecture should be defined based on the architecture framework. The Data Architecture is sometimes referred to as the metamodel of the architecture practice.

2.5 Phase C: Application Architecture

The Application Architecture of the architecture practice defines the functionality required to generate, maintain, publish, distribute, and govern the architecture deliverables as defined in the architecture framework. A key focus should be on the modeling toolsets required for modeling, but it should not be the only focus. Refer to the TOGAF Standard — Architecture Development Method for guidance on selecting a toolset. Publishing the architecture deliverables to address specific views in the architecture framework would sometimes require specialized or customized functionality and should not be neglected.

2.6 Phase D: Technology Architecture

The Technology Architecture of the architecture practice should define technology infrastructure supporting the architecture practice.

2.7 Phase E: Opportunities & Solutions

A critical factor to consider during this phase of planning the establishment of the architecture practice is the organizational change that is required and how this will be achieved.

2.8 Phase F: Migration Planning

The focus should not only be on the Information Systems Architecture components in this phase, but include the Business Architecture. The adoption of the architecture process and framework will have a major impact on the overall establishment of the architecture practice in the organization.

2.9 Phase G: Implementation Governance

The implementation of the Business Architecture of the architecture practice should be the focus of this phase. Changing practices within the organization to adopt a more structured and disciplined approach will be a challenge and should be addressed by the appropriate organizational change techniques.

2.10 Phase H: Architecture Change Management

Changes to the architecture of the architecture practice should be managed by this phase. These changes are usually triggered during the execution of architecture projects. A typical change would be the requirement for a new architecture deliverable. This would impact on all the architecture domains of the architecture practice.

2.11 Requirements Management

Understanding and managing the requirements for the architecture practice is crucial. Requirements should be clearly articulated and align to the architecture practice vision.

Additional guidance on how to construct an Enterprise Architecture Capability can be found in the TOGAF® Series Guide: The TOGAF Leader's Guide to Establishing and Evolving an EA Capability.

Chapter 3: Architecture Governance

This chapter provides a framework and guidelines for Architecture Governance.

3.1 Introduction

This section describes the nature of governance, and the levels of governance.

3.1.1 Levels of Governance within the Enterprise

Architecture Governance is the practice and orientation by which Enterprise Architectures and other architectures are managed and controlled at an enterprise-wide level.

Architecture Governance typically does not operate in isolation, but within a hierarchy of governance structures, which, particularly in the larger enterprise, can include all of the following as distinct domains with their own disciplines and processes:

- Corporate Governance
- Technology Governance
- IT Governance
- Architecture Governance

Each of these domains of governance may exist at multiple geographic levels — global, regional, and local — within the overall enterprise.

Corporate governance is thus a broad topic, beyond the scope of an Enterprise Architecture framework such as the TOGAF framework.

This and related subsections are focused on Architecture Governance; but they describe it in the context of enterprise-wide governance, because of the hierarchy of governance structures within which it typically operates, as explained above.

In particular, this and following sections aim to:

- Provide an overview of the nature of governance as a discipline in its own right
- Describe the governance context in which Architecture Governance typically functions within the enterprise
- Describe a practical Architecture Governance Framework that can be adapted and applied, both for Enterprise Architecture and for other forms of IT architecture

3.1.2 Nature of Governance

3.1.2.1 Governance: A Generic Perspective

Governance is essentially about ensuring that business is conducted properly. It is less about overt control and strict adherence to rules, and more about guidance and effective and equitable usage of resources to ensure sustainability of an organization's strategic objectives.

The following outlines the basic principles of corporate governance, as identified by the Organization for Economic Co-operation and Development (OECD):

- Focuses on the rights, roles, and equitable treatment of shareholders
- Disclosure and transparency and the responsibilities of the board
- Ensures:
 - Sound strategic guidance of the organization
 - Effective monitoring of management by the board
 - Board accountability for the company and to the shareholders
- Responsibilities of the Board:
 - Reviewing and guiding corporate strategy
 - Setting and monitoring achievement of management's performance objectives

Supporting this, the OECD considers a traditional view of governance as: "... the system by which business corporations are directed and controlled. The corporate governance structure specifies the distribution of rights and responsibilities among different participants in the corporation — such as the board, managers, shareholders, and other stakeholders — and spells out the rules and procedures for making decisions on corporate affairs. By doing this, it also provides the structure through which the company objectives are set, and the means of attaining those objectives and monitoring performance" (Source: OECD, 2001).

3.1.2.2 Characteristics of Governance

The following characteristics have been adapted from *Corporate Governance* (Naidoo, 2002) and are positioned here to highlight both the value and necessity for governance as an approach to be adopted within organizations and their dealings with all involved parties:

Discipline	All involved parties will have a commitment to adhere to procedures, processes, and authority structures established by the organization.
Transparency	All actions implemented and their decision support will be available for inspection by authorized organization and provider parties.
Independence	All processes, decision-making, and mechanisms used will be established so as to minimize or avoid potential conflicts of interest.
Accountability	Identifiable groups within the organization — e.g., governance boards who take actions or make decisions — are authorized and accountable for their actions.
Responsibility	Each contracted party is required to act responsibly to the organization and its stakeholders.

Fairness All decisions taken, processes used, and their implementation will not be allowed to create unfair advantage to any one particular party.

3.1.3 Technology Governance

Technology governance controls how an organization utilizes technology in the research, development, and production of its goods and services. Although it may include IT governance activities, it often has a broader scope.

Technology governance is a key capability, requirement, and resource for most organizations because of the pervasiveness of technology across the organizational spectrum.

Recent studies have shown that many organizations have a balance in favor of intangibles rather than tangibles that require management. Given that most of these intangibles are informational and digital assets, it is evident that businesses are becoming more reliant on IT: and the governance of IT — IT governance — is therefore becoming an even more important part of technology governance.

These trends also highlight the dependencies of businesses on not only the information itself but also the processes, systems, and structures that create, deliver, and consume it. As the shift to increasing value through intangibles increases in many industry sectors, so risk management must be considered as key to understanding and moderating new challenges, threats, and opportunities.

Not only are organizations increasingly dependent on IT for their operations and profitability, but also their reputation, brand, and ultimately their values are also dependent on that same information and supporting technology.

3.1.4 IT Governance

IT governance provides the framework and structure that links IT resources and information to enterprise goals and strategies. Furthermore, IT governance institutionalizes best practices for planning, acquiring, implementing, and monitoring IT performance, to ensure that the enterprise's IT assets support its business objectives.

In recent years, IT governance has become integral to the effective governance of the modern enterprise. Businesses are increasingly dependent on IT to support critical business functions and processes; and to successfully gain competitive advantage, businesses need to manage effectively the complex technology that is pervasive throughout the organization in order to respond quickly and safely to business needs.

In addition, regulatory environments around the world are increasingly mandating stricter enterprise control over information, driven by increasing reports of information system disasters and electronic fraud. The management of IT-related risk is now widely accepted as a key part of enterprise governance.

It follows that an IT governance strategy, and an appropriate organization for implementing the strategy, must be established with the backing of executive management, clarifying who owns the enterprise's IT resources, and, in particular, who has ultimate responsibility for their enterprise-wide integration.

3.1.4.1 *An IT Controls Framework — COBIT*

As with corporate governance, IT governance is a broad topic, beyond the scope of an Enterprise Architecture framework such as the TOGAF framework. A good source of detailed information on IT governance is the COBIT® framework (Control Objectives for Information and related Technology). This is an open standard for control over IT, developed and promoted by the IT Governance Institute (ITGI), and published by the Information Systems Audit and Control Foundation (ISACF). COBIT controls may provide useful aides to running a compliance strategy.

3.1.5 **Architecture Governance: Overview**

3.1.5.1 *Architecture Governance 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

3.1.5.2 *Architecture Governance as a Board-Level Responsibility*

This section aims to provide the impetus for opening up IT and Architecture Governance so that the business responsibilities associated with architecture activities and artifacts can be elucidated and managed.

3.1.5.3 *The TOGAF Standard and Architecture Governance*

Phase G of the TOGAF ADM (see the TOGAF Standard — Architecture Development Method) is dedicated to implementation governance, which concerns itself with the realization of the architecture through change projects. Implementation governance is just one aspect of Architecture Governance, which covers the management and control of all aspects of the development and evolution of Enterprise Architectures and other architectures within the enterprise.

Architecture Governance needs to be supported by an Architecture Governance Framework (described in [Section 3.2](#)) which assists in identifying effective processes so that the business responsibilities associated with Architecture Governance can be elucidated, communicated, and managed effectively.

3.2 Architecture Governance Framework

This section describes a conceptual and organizational framework for Architecture Governance

As previously explained, Phase G of the TOGAF ADM (see the TOGAF Standard — Architecture Development Method) is dedicated to implementation governance, which concerns itself with the realization of the architecture through change projects.

Implementation governance is just one aspect of Architecture Governance, which covers the management and control of all aspects of the development and evolution of Enterprise Architectures and other architectures within the enterprise.

Architecture Governance needs to be supported by an Architecture Governance Framework, described below. The governance framework described is a generic framework that can be adapted to the existing governance environment of an enterprise. It is intended to assist in identifying effective processes and organizational structures, so that the business responsibilities associated with Architecture Governance can be elucidated, communicated, and managed effectively.

3.2.1 Architecture Governance Framework — Conceptual Structure

3.2.1.1 Key Concepts

Conceptually, Architecture Governance is an approach, a series of processes, a cultural orientation, and set of owned responsibilities that ensure the integrity and effectiveness of the organization's architectures.

The key concepts are illustrated in [Figure 3-1](#).

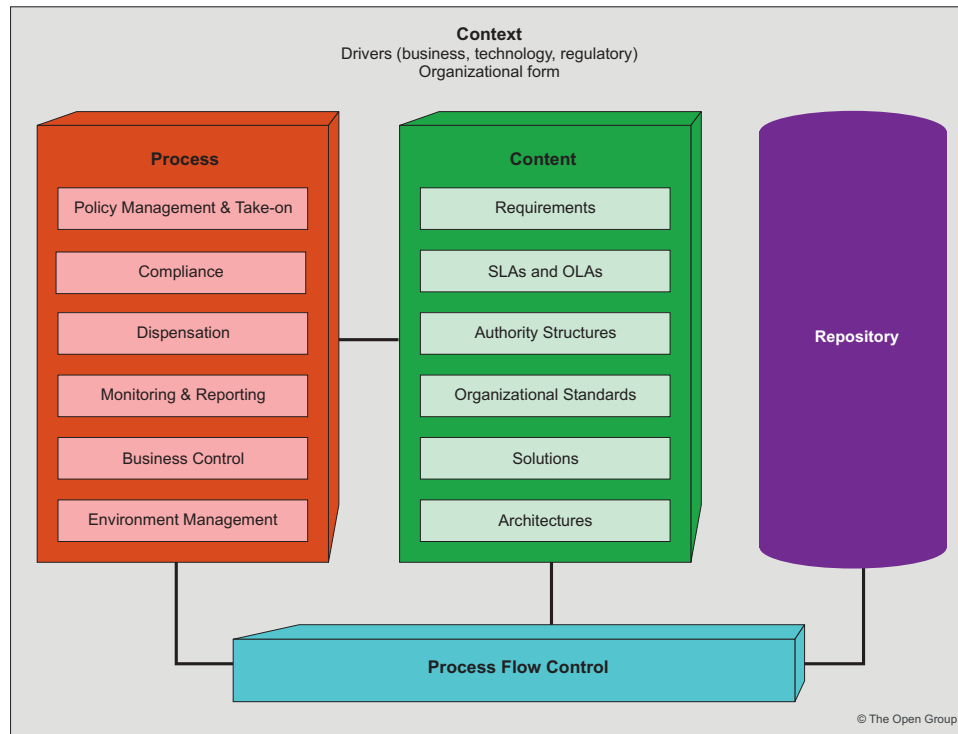


Figure 3-1 Architecture Governance Framework — Conceptual Structure

The split of process, content, and context are key to the support of the Architecture Governance initiative, by allowing the introduction of new governance material (legal, regulatory, standards-based, or legislative) without unduly impacting the processes. This content-agnostic approach ensures that the framework is flexible. The processes are typically independent of the content and implement a proven best practice approach to active governance.

The Architecture Governance Framework is integral to the Enterprise Continuum, and manages all content relevant both to the architecture itself and to Architecture Governance processes.

3.2.1.2 Key Architecture Governance Processes

Governance processes are required to identify, manage, audit, and disseminate all information related to architecture management, contracts, and implementation. These governance processes will be used to ensure that all architecture artifacts and contracts, principles, and Operational-Level Agreements (OLAs) are monitored on an ongoing basis with clear auditability of all decisions made.

Policy Management and Take-On

All architecture amendments, contracts, and supporting information must come under governance through a formal process in order to register, validate, ratify, manage, and publish new or updated content. These processes will ensure the orderly integration with existing governance content such that all relevant parties, documents, contracts, and supporting information are managed and audited.

Compliance

Compliance assessments against Service-Level Agreements (SLAs), OLAs, standards, and regulatory requirements will be implemented on an ongoing basis to ensure stability, conformance, and performance monitoring. These assessments will be reviewed and either accepted or rejected depending on the criteria defined within the governance framework.

Dispensation (also known as Waiver)

A Compliance Assessment can be rejected where the subject area (design, operational, service level, or technology) are not compliant. In this case the subject area can:

1. Be adjusted or realigned in order to meet the compliance requirements
2. Request a dispensation

Where a Compliance Assessment is rejected, an alternate route to meeting interim conformance is provided through dispensations. These are granted for a given time period and a set of identified service and operational criteria that must be enforced during the lifespan of the dispensation. Dispensations are not granted indefinitely, but are used as a mechanism to ensure that service levels and operational levels are met while providing a level of flexibility in their implementation and timing. The time-bound nature of dispensations ensures that they are a major trigger in the compliance cycle.

Monitoring and Reporting

Performance management is required to ensure that both the operational and service elements are managed against an agreed set of criteria. This will include monitoring against SLAs and OLAs, feedback for adjustment, and reporting.

Internal management information will be considered in [Environment Management](#).

Business Control

Business Control relates to the processes invoked to ensure compliance with the organization's business policies.

Environment Management

This identifies all the services required to ensure that the repository-based environment underpinning the governance framework is effective and efficient. This includes the physical and logical repository management, access, communication, training, and accreditation of all users.

All architecture artifacts, service agreements, contracts, and supporting information must come under governance through a formal process in order to register, validate, ratify, manage, and publish new or updated content. These processes will ensure the orderly integration with existing governance content such that all relevant parties, documents, contracts, and supporting information are managed and audited.

The governance environment will have a number of administrative processes defined in order to effect a managed service and process environment. These processes will include user management, internal SLAs (defined in order to control its own processes), and management information reporting.

3.2.2 Architecture Governance Framework — Organizational Structure

3.2.2.1 Overview

Architecture Governance is the practice and orientation by which Enterprise Architectures and other architectures are managed and controlled. In order to ensure that this control is effective within the organization, it is necessary to have the correct organizational structures established to support all governance activities.

An Architecture Governance structure for effectively implementing the approach described in this section will typically include the following levels, which may in practice involve a combination of existing IT governance processes, organizational structures, and capabilities. They will typically include the following:

- Global governance board
- Local governance board
- Design authorities
- Working parties

The architecture organization illustrated in [Figure 3-2](#) highlights the major structural elements required for an Architecture Governance initiative. While each enterprise will have differing requirements, it is expected that the basics of the organizational design shown in [Figure 3-2](#) will be applicable and implementable in a wide variety of organizational types.

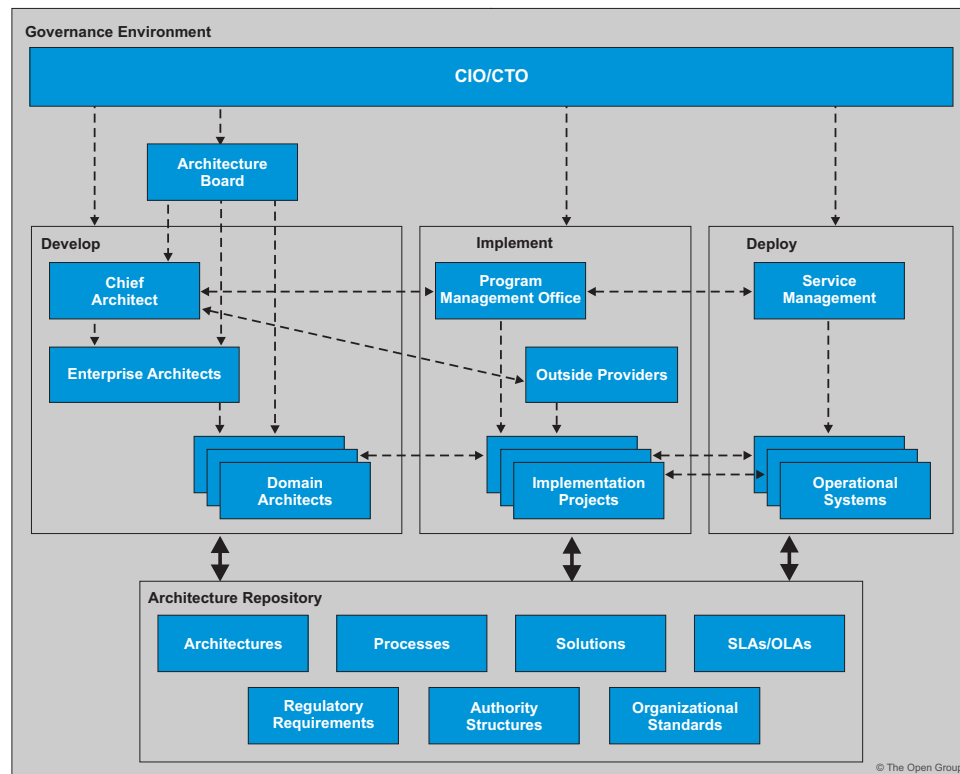


Figure 3-2 Architecture Governance Framework — Organizational Structure

3.2.2.2 Key Areas

Figure 3-2 identifies three key areas of architecture management: Develop, Implement, and Deploy. Each of these is the responsibility of one or more groups within the organization, while the Enterprise Continuum is shown to support all activities and artifacts associated with the governance of the architectures throughout their lifecycle.

The Develop responsibilities, processes, and structures are usually linked to the TOGAF ADM and its usage, while the Implement responsibilities, processes, and structures are typically linked to Phase G (see the TOGAF Standard — Architecture Development Method).

As mentioned above, the Architecture Governance Framework is integral to the Enterprise Continuum, and manages all content relevant both to the architectures themselves and to Architecture Governance processes.

3.2.2.3 Operational Benefits

As illustrated in Figure 3-2, the governance of the organization's architectures provides not only direct control and guidance of their development and implementation, but also extends into the operations of the implemented architectures.

The following benefits have been found to be derived through the continuing governance of architectures:

- Links IT processes, resources, and information to organizational strategies and objectives
- Integrates and institutionalizes IT best practices
- Aligns with industry frameworks such as COBIT (planning and organizing, acquiring and implementing, delivering and supporting, and monitoring IT performance)
- Enables the organization to take full advantage of its information, infrastructure, and hardware and software assets
- Protects the underlying digital assets of the organization
- Supports regulatory and best practice requirements such as auditability, security, responsibility, and accountability
- Promotes visible risk management

These benefits position the TOGAF Architecture Governance Framework as an approach, a series of processes, a cultural orientation, and a set of owned responsibilities, that together ensure the integrity and effectiveness of the organization's architectures.

3.3 Architecture Governance in Practice

This section provides practical guidelines for the effective implementation of Architecture Governance

3.3.1 Architecture Governance — Key Success Factors

It is important to consider the following to ensure a successful approach to Architecture Governance, and to the effective management of the Architecture Contract:

- 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

3.3.2 Elements of an Effective Architecture Governance Strategy

3.3.2.1 Architecture Governance and Corporate Politics

An Enterprise Architecture imposed without appropriate executive support and corporate political alignment is bound to fail. In order to succeed, the Enterprise Architecture must reflect the needs of the organization. Enterprise Architects, if they are not involved in the development of business strategy, must at least have a fundamental understanding of it and of the prevailing business issues facing the organization. It may even be necessary for them to be involved in the system deployment process and to ultimately own the investment and product selection decisions arising from the implementation of the Technology Architecture.

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 form an integral part of any effective Architecture Governance strategy.

- A cross-organizational Architecture Board (see [Chapter 4](#)) must be established with the backing of executive management to oversee the implementation of the Enterprise Architecture Governance strategy
- A comprehensive set of Architecture Principles (see the TOGAF Standard — ADM Techniques) 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 [Chapter 6](#)) 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

Chapter 4: Architecture Board

This chapter provides guidelines for establishing and operating an Enterprise Architecture Board.

4.1 Role

A key element in a successful Architecture Governance strategy (see [Chapter 3](#)) is a cross-organization Architecture Board to oversee the implementation of the strategy. This body should be representative of all the key stakeholders in the architecture, and will typically comprise a group of executives responsible for the review and maintenance of the overall architecture.

Architecture Boards may have global, regional, or business line scope. Particularly in larger enterprises, Architecture Boards typically comprise representatives from the organization at a minimum of two levels:

- Local (domain experts, line responsibility)
- Global (organization-wide responsibility)

In such cases, each board will be established with identifiable and articulated:

- Responsibilities and decision-making capabilities
- Remit and authority limits

4.2 Responsibilities

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 the 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
- Ensuring that the discipline of architecture-based development is adopted

- Supporting a visible escalation capability for out-of-bounds decisions

Further responsibilities from an operational perspective should include:

- All aspects of monitoring and control of the Architecture Contract
- Meeting on a regular basis
- Ensuring the effective and consistent management and implementation of the architectures
- Resolving ambiguities, issues, or conflicts that have been escalated
- Providing advice, guidance, and information
- Ensuring compliance with the architectures, and granting dispensations that are in keeping with the technology strategy and objectives
- Considering policy (schedule, SLAs, etc.) changes where similar dispensations are requested and granted; e.g., new form of service requirement
- Ensuring that all information relevant to the implementation of the Architecture Contract is published under controlled conditions and made available to authorized parties
- Validation of reported service levels, cost savings, etc.

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
- Providing a fundamental control mechanism for ensuring the effective implementation of the architecture
- Establishing and maintaining the link between the implementation of the architecture, the architectural strategy and objectives embodied in the Enterprise Architecture, and the strategic objectives of the business
- Identifying divergence from the architecture and planning activities for realignment through dispensations or policy updates

4.3 Setting Up the Architecture Board

4.3.1 Triggers

One or more of the following occurrences typically triggers the establishment of an Architecture Board:

- New CIO
- Merger or acquisition
- Organizational restructuring
- Consideration of a move to newer forms of computing
- Recognition that IT is poorly aligned to business

- Desire to achieve competitive advantage via technology
- Creation of an Enterprise Architecture program
- Significant business change or rapid growth
- Requirement for complex, cross-functional solutions

In many companies, the executive sponsor of the initial architecture effort is the CIO (or other senior executive). However, to gain broad corporate support, a sponsoring body has more influence. This sponsoring body is here called an Architecture Board, but the title is not important. Whatever the name, it is the executive-level group responsible for the review and maintenance of the strategic architecture and all of its sub-architectures.

The Architecture Board is the sponsor of the architecture within the enterprise, but the Architecture Board itself needs an executive sponsor from the highest level of the corporation. This commitment must span the planning process and continue into the maintenance phase of the architecture project. In many companies that fail in an architecture planning effort, there is a notable lack of executive participation and encouragement for the project.

A frequently overlooked source of Architecture Board members is the company's Board of Directors. These individuals invariably have diverse knowledge about the business and its competition. Because they have a significant impact on the business vision and objectives, they may be successful in validating the alignment of IT strategies to business objectives.

4.3.2 Size of the Board

The recommended size for an Architecture Board is four or five (and no more than ten) permanent members. In order to keep the Architecture Board to a reasonable size, while ensuring enterprise-wide representation on it over time, membership of the Architecture Board may be rotated, giving decision-making privileges and responsibilities to various senior managers. This may be required in any case, due to some Architecture Board members finding that time constraints prevent long-term active participation.

An organization may set up its Architecture Board using representational means, such that each Board member is assigned a set of stakeholders to represent in the meetings. It is then incumbent upon Board members to meet with stakeholders to gain their views on the various agenda items.

However, some continuity must exist on the Architecture Board, to prevent the corporate architecture from varying from one set of ideas to another. One technique for ensuring rotation with continuity is to have set terms for the members, and to have the terms expire at different times.

In the ongoing architecture process following the initial architecture effort, the Architecture Board may be re-chartered. The executive sponsor will normally review the work of the Architecture Board and evaluate its effectiveness; if necessary, the Architecture Compliance review process is updated or changed.

4.3.3 Board Structure

The TOGAF Architecture Governance Framework (see [Section 3.2](#)) provides a generic organizational framework that positions the Architecture Board in the context of the broader governance structures of the enterprise. This structure identifies the major organizational groups and responsibilities, as well as the relationship between each group. This is a best practice structure, and may be subject to change depending on the organization's form and existing structures.

Consideration must be given to the size of the organization, its form, and how the IT functions are implemented. This will provide the basis for designing the Architecture Board structure within the context of the overall governance environment. In particular, consideration should be given to the concept of global ownership and local implementation, and the integration of new concepts and technologies from all areas implementing against architectures.

The structure of the Architecture Board should reflect the form of the organization. The Architecture Governance structure required may well go beyond the generic structures outlined in the TOGAF Architecture Governance Framework (see [Section 3.2](#)). The organization may need to define a combination of the IT governance process in place and the existing organizational structures and capabilities, which typically include the following types of body:

- Global governance board
- Local governance board
- Design authorities
- Working parties

4.4 Operation of the Architecture Board

This section describes the operation of the Architecture Board particularly from the governance perspective.

4.4.1 General

Architecture Board meetings should be conducted within clearly identified agendas with explicit objectives, content coverage, and defined actions. In general, board meetings will be aligned with best practice, such as given in the COBIT framework (see [Section 3.1.4.1](#)).

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
- Establishing and maintaining the link between the implementation of the architectures and the stated strategy and objectives of the organization (business and IT)

- Identifying divergence from the contract and planning activities to realign with the contract through dispensations or policy updates

4.4.2 Preparation

Each participant will receive an agenda and any supporting documentation — e.g., dispensation requests, performance management reports, etc. — and will be expected to be familiar with the contents of each.

Where actions have been allocated to an individual, it is that person's responsibility to report on progress against these.

Each participant must confirm their availability and attendance at the Architecture Board meeting.

4.4.3 Agenda

This section outlines the contents of an Architecture Board meeting agenda. Each agenda item is described in terms of its content only.

Minutes of Previous Meeting

Minutes contain the details of previous Architecture Board meetings as per standard organizational protocol.

Requests for Change

Items under this heading are normally change requests for amendments to architectures, principles, etc., but may also include business control with regard to Architecture Contracts; e.g., ensure that voice traffic to premium numbers, such as weather reports, is barred and data traffic to certain websites is controlled.

Any request for change is made within agreed authority levels and parameters defined by the Architecture Contract.

Dispensations

A dispensation is used as the mechanism to request a change to the existing architectures, contracts, principles, etc. outside of normal operating parameters; e.g., exclude provision of service to a subsidiary, request for unusual service levels for specific business reasons, deploy non-standard technology or products to support specific business initiatives.

Dispensations are granted for a given time period and a set of identified services and operational criteria that must be enforced during the lifespan of the dispensation. Dispensations are not granted indefinitely, but are used as a mechanism to ensure that service levels and operational levels, etc. are met while providing a level flexibility in their implementation and timing. The time-bound nature of dispensations ensures that they are a trigger to the Architecture Compliance activity.

Compliance Assessments

Compliance is assessed against SLAs, Operational-Level Agreements (OLAs), cost targets, and required architecture refreshes. These assessments will be reviewed and either accepted or rejected depending on the criteria defined within the Architecture Governance Framework. The Architecture Compliance assessment report will include details as described.

Dispute Resolution

Disputes that have not been resolved through the Architecture Compliance and dispensation processes are identified here for further action and are documented through the Architecture Compliance assessments and dispensation documentation.

Architecture Strategy and Direction Documentation

This describes the architecture strategies, direction, and priorities and will only be formulated by the global Architecture Board. It should take the form of standard architecture documentation.

Actions Assigned

This is a report on the actions assigned at previous Architecture Board meetings. An action tracker is used to document and keep the status of all actions assigned during the Architecture Board meetings and should consist of at least the following information:

- Reference
- Priority
- Action description
- Action owner
- Action details
- Date raised
- Due date
- Status
- Type
- Resolution date

Contract Documentation Management

This is a formal acceptance of updates and changes to architecture documentation for onward publication.

Any Other Business (AOB)

Description of issues not directly covered under any of the above. These may not be described in the agenda but should be raised at the beginning of the meeting. Any supporting documentation must be managed as per all Architecture Governance documentation.

Schedule of Meetings

All meeting dates should be detailed and published.

Chapter 5: Architecture Contracts

This chapter provides guidelines for defining and using Architecture Contracts.

5.1 Role

Architecture Contracts are the joint agreements between development partners and sponsors on the deliverables, quality, and fitness-for-purpose of an architecture. Successful implementation of these agreements will be delivered through effective Architecture Governance (see [Chapter 3](#)). By implementing a governed approach to the management of contracts, the following will be ensured:

- A system of continuous monitoring to check integrity, changes, decision-making, and audit of all architecture-related activities within the organization
- Adherence to the principles, standards, and requirements of the existing or developing architectures
- Identification of risks in all aspects of the development and implementation of the architecture(s) covering the internal development against accepted standards, policies, technologies, and products as well as the operational aspects of the architectures such that the organization can continue its business within a resilient environment
- A set of processes and practices that ensure accountability, responsibility, and discipline with regard to the development and usage of all architectural artifacts
- A formal understanding of the governance organization responsible for the contract, their level of authority, and scope of the architecture under the governance of this body

The traditional Architecture Contract is an agreement between the sponsor and the architecture function or Information Systems (IS) department. However, increasingly more services are being provided by systems integrators, applications providers, and service providers, co-ordinated through the architecture function or IS department. There is therefore a need for an Architecture Contract to establish joint agreements between all parties involved in the architecture development and delivery.

Architecture Contracts may occur at various stages of the ADM; for example:

- The Statement of Architecture Work created in Phase A of the TOGAF Standard — Architecture Development Method is effectively an Architecture Contract between the architecting organization and the sponsor of the Enterprise Architecture (or the IT governance function)
- 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

Each of these arrangements will normally be governed by an Architecture Contract that

defines the deliverables, quality, and fitness-for-purpose of the developed architecture, and the processes by which the partners in the architecture development will work together.

- 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; typically, this will be either the in-house systems development function, or a major contractor to whom the work is outsourced
 - What is being "implemented" in Phase G of the ADM is the overall Enterprise Architecture

This will typically include the technology infrastructure (from Phase D), and also those enterprise applications and data management capabilities that have been defined in the Application Architecture and Data Architecture (from Phase C), either because they are enterprise-wide in scope, or because they are strategic in business terms, and therefore of enterprise-wide importance and visibility. However, it will typically not include non-strategic business applications, which business units will subsequently deploy on top of the technology infrastructure that is implemented as part of the Enterprise Architecture.
 - In larger-scale implementations, there may well be one Architecture Contract per implementation team in a program of implementation projects
- When the finalized Architecture Definition Document is available (at the end of Phase F), an Architecture Contract will normally be drawn up between the architecting function (or the IT governance function, subsuming the architecting function) and the business users who will subsequently be building and deploying application systems in the architected environment

It is important to bear in mind in all these cases that the ultimate goal is not just an Enterprise Architecture, but a dynamic Enterprise Architecture; i.e., one that allows for flexible evolution in response to changing technology and business drivers, without unnecessary constraints. The Architecture Contract is crucial to enabling a dynamic Enterprise Architecture and is key to governing the implementation.

Typical contents of these three kinds of Architecture Contract are explained below.

5.2 Contents

5.2.1 Statement of Architecture Work

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 (or the IT governance function, on behalf of the enterprise).

The typical contents of a Statement of Architecture Work are as defined in the TOGAF Standard — Architecture Content.

5.2.2 Contract between Architecture Design and Development Partners

This is a signed statement of intent on designing and developing the Enterprise Architecture, or significant parts of it, from partner organizations, including systems integrators, applications providers, and service providers.

Increasingly, the development of one or more architecture domains (Business, Data, Application, Technology) may be contracted out, with the enterprise's architecture function providing oversight of the overall Enterprise Architecture, and co-ordination and control of the overall effort. In some cases even this oversight role may be contracted out, although most enterprises prefer to retain that core responsibility in-house.

Whatever the specifics of the contracting-out arrangements, the arrangements themselves will normally be governed by an Architecture Contract that defines the deliverables, quality, and fitness-for-purpose of the developed architecture, and the processes by which the partners in the architecture development will work together.

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
- Conformance requirements
- Architecture development and management process and roles
- Target architecture measures
- Defined phases of deliverables
- Prioritized joint workplan
- Time window(s)
- Architecture delivery and business metrics

The template for this contract will normally be defined as part of the Preliminary Phase of the ADM, if not existing already, and the specific contract will be defined at the appropriate stage of the ADM, depending on the particular work that is being contracted out.

5.2.3 Contract between Architecting Function and Business Stakeholders

When the Implementation and Migration Plan has been agreed (at the end of Phase F), an Architecture Contract may be drawn up between the architecting function and the business stakeholders who will subsequently be building and deploying the architected business solutions.

A business stakeholder's Architecture Contract may include:

- Introduction and background
- The nature of the agreement
- Scope
- Strategic requirements
- Architecture deliverables that meet the business requirements
- Conformance requirements
- Architecture adopters
- Time window
- Architecture business metrics
- SLA

This contract is also used to manage changes to the Enterprise Architecture in Phase H.

5.3 Relationship to Architecture Governance

The Architecture Contract document produced in Phase G of the ADM figures prominently in the area of Architecture Governance, as explained in [Chapter 3](#).

In the context of Architecture Governance, the Architecture Contract is often used as a means of driving architecture change.

In order to ensure that the Architecture Contract is effective and efficient, the following aspects of the governance framework may need to be introduced into Phase G:

- Simple processes
- People-centered authority
- Strong communication
- Timely responses and an effective escalation process
- Supporting organizational structures
- Status tracking of architecture implementation

Chapter 6: Architecture Compliance

This chapter provides guidelines for ensuring project compliance to the architecture.

6.1 Introduction

Ensuring the compliance of individual projects with the Enterprise Architecture is an essential aspect of Architecture Governance (see [Chapter 3](#)). To this end, the 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 **Enterprise and IT Governance** functions will define a formal Architecture Compliance review process (see [Section 6.3](#)) for reviewing the compliance of all projects to the Enterprise Architecture

Apart from defining formal processes, the Architecture Governance function (see [Chapter 3](#)) may also stipulate that the architecture function should extend beyond the role of architecture definition and standards selection, and participate also in the technology selection process, and even in the commercial relationships involved in external service provision and product purchases. This may help to minimize the opportunity for misinterpretation of the Enterprise Architecture, and maximize the value of centralized commercial negotiation.

6.2 Terminology: The Meaning of Architecture Compliance

A key relationship between the architecture and the implementation lies in the definitions of the terms "conformant", "compliant", etc. While terminology usage may differ between organizations, the concepts of levels of conformance illustrated in [Figure 6-1](#) should prove useful in formulating an IT compliance strategy.

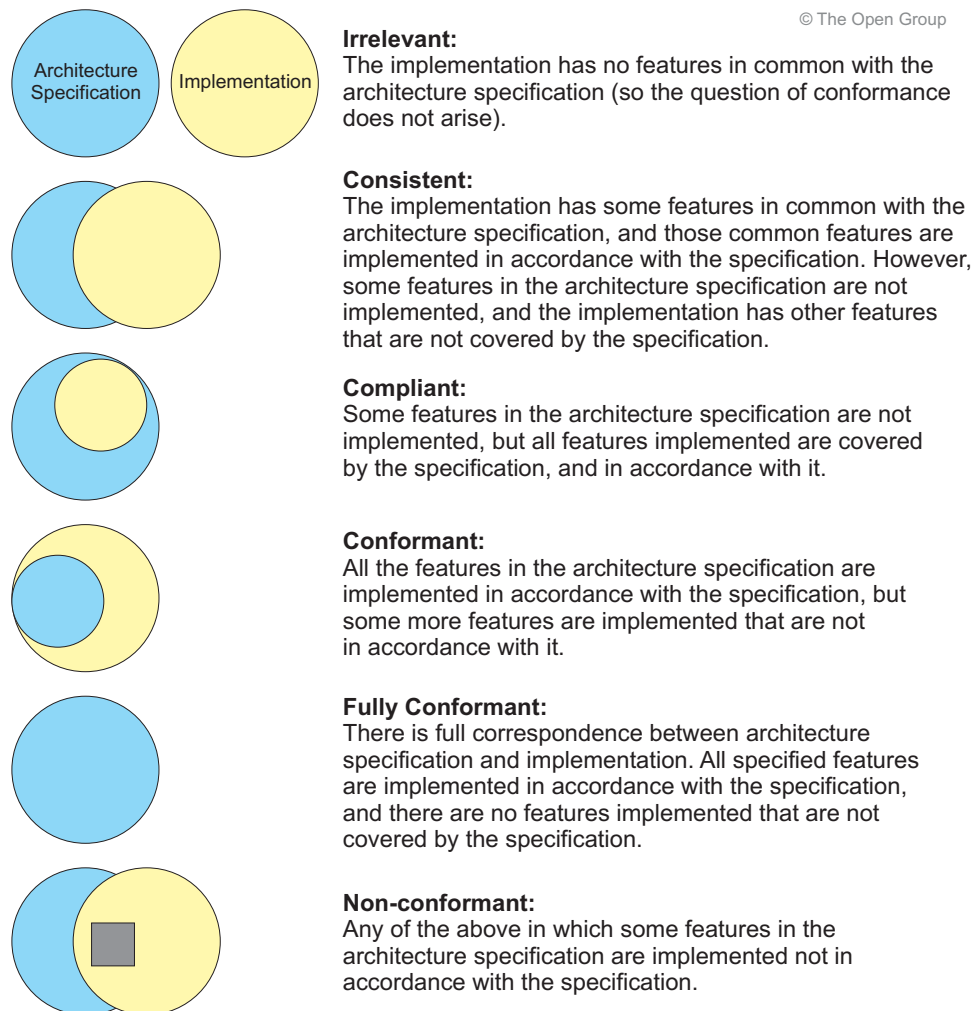


Figure 6-1 Levels of Architecture Conformance

The phrase "in accordance with" in [Figure 6-1](#) means:

- Supports the stated strategy and future directions
- Adheres to the stated standards (including syntax and semantic rules specified)
- Provides the stated functionality
- Adheres to the stated principles; for example:
 - Open wherever possible and appropriate
 - Re-use of component building blocks wherever possible and appropriate

6.3 Architecture Compliance Reviews

An Architecture Compliance review is a scrutiny of the compliance of a specific project against established architectural criteria, spirit, and business objectives. A formal process for such reviews normally forms the core of an Enterprise Architecture Compliance strategy.

6.3.1 Purpose

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

This in turn means that the overall project time is shortened, and that the business gets the bottom-line benefit of the architecture development faster.
- Ensure the application of best practices to architecture work
- Provide an overview of the compliance of an architecture to mandated enterprise standards
- Identify where the standards themselves may require modification
- Identify services that are currently application-specific but might be provided as part of the enterprise infrastructure
- Document strategies for collaboration, resource sharing, and other synergies across multiple architecture teams
- Take advantage of advances in technology
- Communicate to management the status of business and technical readiness of the project
- Identify key criteria for procurement activities; e.g., for inclusion in Commercial Off-The-Shelf (COTS) product Request for Information (RFI)/Request for Proposal (RFP) documents
- Identify and communicate significant architectural gaps to product and service providers

Apart from the generic goals related to quality assurance outlined above, there are additional, more politically-oriented motivations for conducting Architecture Compliance reviews, which may be relevant in particular cases:

- 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, as opposed to what is technically more pleasing or elegant
- The output of the Architecture Compliance review is one of the few measurable deliverables to the executive management to assist in decision-making
- Architecture reviews can serve as a way for the architecture organization to engage with development projects that might otherwise proceed without involvement of the architecture function
- Architecture reviews can demonstrate rapid and positive support to the enterprise business community:

- The Enterprise Architecture and Architecture Compliance helps ensure the alignment of IT projects with business objectives
- Architects can sometimes be regarded as being deep into technical infrastructure and far removed from the core business
- Since an Architecture Compliance review tends to look primarily at the critical risk areas of a system, it often highlights the main risks for system owners

While compliance to architecture is required for development and implementation, non-compliance also provides a mechanism for highlighting:

- Areas to be addressed for realignment
- Areas for consideration for integration into the architectures as they are uncovered by the compliance processes

The latter point identifies the ongoing change and adaptability of the architectures to requirements that may be driven by indiscipline, but also allows for changes to be registered by faster moving changes in the operational environment. Typically, dispensations (see [Section 3.1.4](#)) will be used to highlight these changes and set in motion a process for registering, monitoring, and assessing the suitability of any changes required.

6.3.2 Timing

Timing of compliance activities should be considered with regard to the development of the architectures themselves.

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)

Architecture project timings for assessments should include:

- Project initiation
- Initial design
- Major design changes
- *Ad hoc*

The Architecture Compliance review is typically targeted for a point in time when business requirements and the Enterprise Architecture are reasonably firm, and the project architecture is taking shape, well before its completion.

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, with the obvious proviso that there needs to have been some significant development of the project architecture in order for there to be something to review.

Inputs to the Architecture Compliance review may come from other parts of the standard project lifecycle, which may have an impact on timing.

6.3.3 Governance and Personnel Scenarios

In terms of the governance and conduct of the Architecture Compliance review, and the personnel involved, there are various possible scenarios:

- For smaller-scale projects, the review process could simply take the form of a series of questions that the project architects or project leaders pose to themselves, using the checklists provided below, perhaps collating the answers into some form of project report to management

The need to conduct such a process is normally included in overall enterprise-wide governance policies.

- Where the project under review has not involved a practicing or full-time architect to date (for example, in an application-level project), the purpose of the review is typically to bring to bear the architectural expertise of an Enterprise Architecture function

In such a case, the Enterprise Architecture function would be organizing, leading, and conducting the review, with the involvement of business domain experts. In such a scenario, the review is not a substitute for the involvement of architects in a project, but it can be a supplement or a guide to their involvement. It is probable that a database will be necessary to manage the volume of data that would be produced in the analysis of a large system or set of systems.

- In most cases, particularly in larger-scale projects, the architecture function will have been deeply involved in, and perhaps leading, the development project under review

(This is the typical TOGAF scenario.) In such cases, the review will be co-ordinated by the lead Enterprise Architect, who will assemble a team of business and technical domain experts for the review, and compile the answers to the questions posed during the review into some form of report. The questions will typically be posed during the review by the business and technical domain experts. Alternatively, the review might be led by a representative of an Architecture Board or some similar body with enterprise-wide responsibilities.

In all cases, the Architecture Compliance review process needs the backing of senior management, and will typically be mandated as part of corporate Architecture Governance policies (see [Chapter 3](#)). Normally, the enterprise CIO or Enterprise Architecture Board (see [Chapter 4](#)) will mandate architecture reviews for all major projects, with subsequent annual reviews.

6.4 Architecture Compliance Review Process

6.4.1 Overview

The Architecture Compliance review process is illustrated in [Figure 6-2](#).

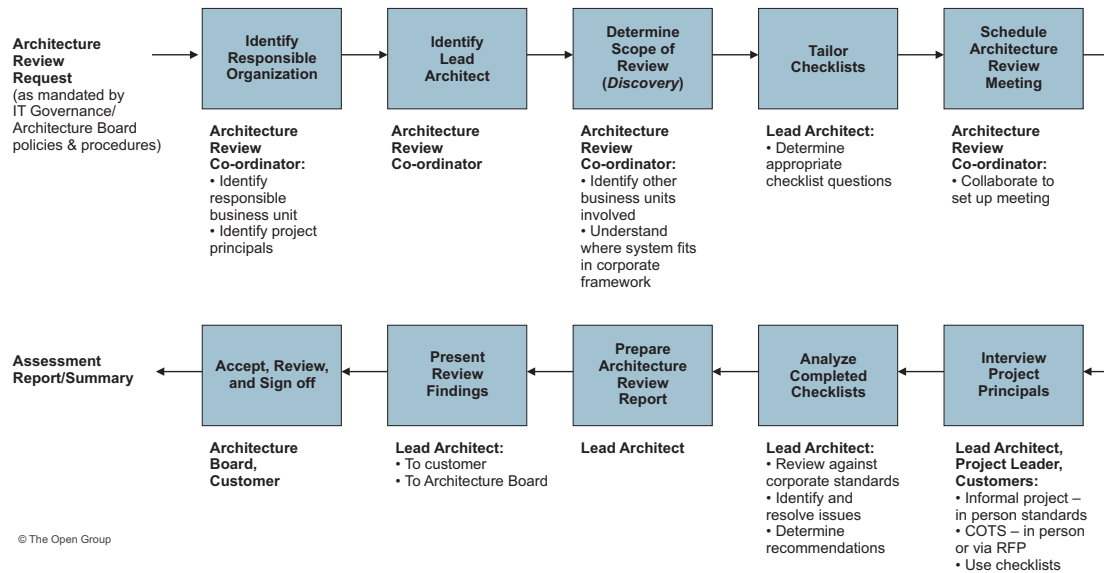


Figure 6-2 Architecture Compliance Review Process

Note: Another way to gain conformance checks in architecture is through the use of traceability in models and "drill-down" diagrams. In this technique, the Enterprise Architect creates a top-level view of an architecture which the Solution Architects further refine by tracing their elements to the Enterprise Architecture model. These refinements can be managed through so-called "drill-down" diagrams which contain all the refinement elements and their traceability. In this way, conformance can be reviewed and at least partially automated by tools (such as having a script that checks that all Enterprise Architecture elements have refinements in the Solution Architecture).

6.4.2 Roles

The main roles in the process are tabulated below.

No.	Role	Responsibilities	Notes
1	Architecture Board	To ensure that Enterprise 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 business and technically coherent and future-proof.	An appropriate architecture specialist.
5	Architect	One of the Lead Enterprise Architect's 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 Enterprise 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.

6.4.3 Steps

The main steps in the process are tabulated below.

No.	Action	Notes	Who
1	Request architecture review.	As mandated by 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: <ul style="list-style-type: none"> ■ 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.	<ul style="list-style-type: none"> ■ To Customer ■ To Architecture Board 	Lead Enterprise Architect
11	Accept review and sign off.		Architecture Board and Customer

No.	Action	Notes	Who
12	Send assessment report/summary to Architecture Review Co-ordinator.		Lead Enterprise Architect

6.5 Architecture Compliance Review Checklists

The following review checklists provide a wide range of typical questions that may be used in conducting Architecture Compliance reviews, relating to various aspects of the architecture. The organization of the questions includes the basic disciplines of system engineering, information management, security, and systems management. The checklists are based on material provided by a member of The Open Group, and are specific to that organization. Other organizations could use the following checklists with other questions tailored to their own particular needs.

The checklists provided contain too many questions for any single review: they are intended to be tailored selectively to the project concerned (see [Section 6.6](#)). The checklists actually used will typically be developed/selected by subject matter experts. They are intended to be updated annually by interest groups in those areas.

Some of the checklists include a brief description of the Architecture Principle that provokes the question, and a brief description of what to look for in the answer. These extensions to the checklist are intended to allow the intelligent re-phrasing of the questions, and to give the user of the checklist a feel for why the question is being asked.

Occasionally the questions will be written, as in RFPs, or in working with a senior project architect. More typically they are expressed orally, as part of an interview or working session with the project.

The checklists provided here are designed for use in individual architecture projects, not for business domain architecture or for architecture across multiple projects. (Doing an architecture review for a larger sphere of activity, across multiple business processes and system projects, would involve a similar process, but the checklist categories and their contents would be different.)

6.5.1 Hardware and Operating System Checklist

1. What is the project's lifecycle approach?
2. At what stage is the project in its lifecycle?
3. What key issues have been identified or analyzed that the project believes will drive evaluations of hardware and operating systems for networks, servers, and end-user devices?
4. What system capabilities will involve high-volume and/or high-frequency data transfers?
5. How does the system design impact or involve end-user devices?
6. What is the quantity and distribution (regional and global) of usage, data storage, and processing?

7. What applications are affinitized with your project by similarities in data, application services, etc.? To what degree is data affinitized with your project?
8. What hardware and operating system choices have been made before the functional design of key elements of the system?
9. If hardware and operating system decisions were made outside of the project's control:
 - What awareness does the project have of the rationale for those decisions?
 - How can the project influence those decisions as system design takes shape?
10. If some non-standards have been chosen:
 - What are the essential business and technical requirements for not using corporate standards?
 - Is this supported by a business case?
 - Have the assumptions in the business case been subject to scrutiny?
11. What is your process for evaluating full lifecycle costs of hardware and operating systems?
12. How has corporate financial management been engaged in evaluation of lifecycle costs?
13. Have you performed a financial analysis of the supplier?
14. Have you made commitments to any supplier?
15. Do you believe your requirements can be met by only one supplier?

6.5.2 Software Services and Middleware Checklist

1. Describe how error conditions are defined, raised, and propagated between application components.
2. Describe the general pattern of how methods are defined and arranged in various application modules.
3. Describe the general pattern for how method parameters are defined and organized in various application modules. Are [in], [in/out], [out] parameters always specified in the same order? Do Boolean values returned by modules have a consistent outcome?
4. Describe the approach that is used to minimize the number of round-trips between client and server calls, particularly for out-of-process calls, and when complex data structures are involved.
5. Describe the major data structures that are passed between major system components.
6. Describe the major communication protocols that are used between major system components.
7. Describe the marshaling techniques that are used between various system components. Describe any specialized marshaling arrangements that are used.
8. Describe to what extent the system is designed with stateful and stateless components.

9. Describe how and when state is saved for both stateful and stateless components.
10. Describe the extent to which objects are created, used, and destroyed *versus* re-used through object pooling.
11. Describe the extent to which the system relies on threading or critical section coding.
12. Describe the approach and the internal documentation that is used internally in the system to document the methods, methods arguments, and method functionality.
13. Describe the code review process that was used to build the system.
14. Describe the unit testing that has been used to test the system components.
15. Describe the pre- and post-condition testing that is included in various system modules.
16. Describe the assertion testing that is included with the system.
17. Do components support all the interface types they need to support or are certain assumptions made about what types of components will call other components either in terms of language bindings or other forms of marshaling?
18. Describe the extent to which big-endian or little-endian data format problems need to be handled across different platforms.
19. Describe if numbers or strings need to be handled differently across different platforms.
20. Describe whether the software needs to check for floating-point round-off errors.
21. Describe how time and date functions manage dates so as to avoid improper handling of time and date calculation or display.
22. Describe what tools or processes have been used to test the system for memory leaks, reachability, or general robustness.
23. Describe the layering of the systems services software. Describe the general number of links between major system components. Is the system composed of a lot of point-to-point interfaces or are major messaging backbones used instead?
24. Describe to what extent the system components are either loosely coupled or tightly coupled.
25. What requirements does the system need from the infrastructure in terms of shared libraries, support for communication protocols, load balancing, transaction processing, system monitoring, naming services, or other infrastructure services?
26. Describe how the system and system components are designed for refactoring.
27. Describe how the system or system components rely on common messaging infrastructure *versus* a unique point-to-point communication structure.

6.5.3 Applications Checklists

6.5.3.1 Infrastructure (Enterprise Productivity) Applications

1. Is there need for capabilities that are not provided through the enterprise's standard infrastructure application products? For example:
 - Collaboration
 - Application sharing
 - Video conferencing
 - Calendaring
 - Email
 - Workflow management
 - Publishing/word processing applications
 - HTML
 - SGML and XML
 - Portable document format
 - Document processing (proprietary format)
 - Desktop publishing
 - Spreadsheet applications
 - Presentation applications
 - Business presentations
 - Image
 - Animation
 - Video
 - Sound
 - CBT
 - Web browsers
 - Data management applications
 - Database interface
 - Document management
 - Product data management
 - Data warehouses/mart
 - Program management applications
 - Project management
 - Program visibility

2. Describe the business requirements for enterprise infrastructure application capabilities that are not met by the standard products.

6.5.3.2 Business Applications

1. Are any of the capabilities required provided by standard products supporting one or more line-of-business applications? For example:
 - Business acquisition applications
 - Sales and marketing
 - Engineering applications
 - Computer-aided design
 - Computer-aided engineering
 - Mathematical and statistics analysis
 - Supplier management applications
 - Supply chain management
 - Customer relationship management
 - Manufacturing applications
 - Enterprise Resource Planning (ERP) applications
 - Manufacturing execution systems
 - Manufacturing quality
 - Manufacturing process engineering
 - Machine and adaptive control
 - Customer support applications
 - Airline logistics support
 - Maintenance engineering
 - Finance applications
 - People applications
 - Facilities applications
 - Information systems applications
 - Systems engineering
 - Software engineering
 - Web developer tools
 - Integrated development environments
 - Lifecycle categories
 - Functional categories

- Specialty categories
 - Computer-aided manufacturing
 - e-Business enablement
 - Business process engineering
 - Statistical quality control
2. Describe the process requirements for business application capabilities that are not met by the standard products.

6.5.3.3 *Application Integration Approach*

1. What integration points (business process/activity, application, data, computing environment) are targeted by this architecture?
2. What application integration techniques will be applied (common business objects [Object Request Brokers (ORBs)], standard data definitions [XML, etc.], common user interface presentation/desktop)?

6.5.4 **Information Management Checklists**

6.5.4.1 *Data Values*

1. What are the processes that standardize the management and use of the data?
2. What business process supports the entry and validation of the data? Use of the data?
3. What business actions correspond to the creation and modification of the data?
4. What business actions correspond to the deletion of the data and is it considered part of a business record?
5. What are the data quality requirements required by the business user?
6. What processes are in place to support data referential integrity and/or normalization?

6.5.4.2 *Data Definition*

1. What are the data model, data definitions, structure, and hosting options of purchased applications (COTS)?
2. What are the rules for defining and maintaining the data requirements and designs for all components of the information system?
3. What shareable repository is used to capture the model content and the supporting information for data?
4. What is the physical data model definition (derived from logical data models) used to design the database?
5. What software development and data management tools have been selected?
6. What data owners have been identified to be responsible for common data definitions, eliminating unplanned redundancy, providing consistently reliable, timely, and accurate information, and protecting data from misuse and destruction?

6.5.4.3 Security/Protection

1. What are the data entity and attribute access rules which protect the data from unintentional and unauthorized alterations, disclosure, and distribution?
2. What are the data protection mechanisms to protect data from unauthorized external access?
3. What are the data protection mechanisms to control access to data from external sources that temporarily have internal residence within the enterprise?

6.5.4.4 Hosting, Data Types, and Sharing

1. What is the discipline for managing sole-authority data as one logical source with defined updating rules for physical data residing on different platforms?
2. What is the discipline for managing replicated data, which is derived from operational sole-authority data?
3. What tier data server has been identified for the storage of high or medium-critical operational data?
4. What tier data server has been identified for the storage of type C operational data?
5. What tier data server has been identified for the storage of decision support data contained in a data warehouse?
6. What Database Management Systems (DBMSs) have been implemented?

6.5.4.5 Common Services

1. What are the standardized distributed data management services (e.g., validation, consistency checks, data edits, encryption, and transaction management) and where do they reside?

6.5.4.6 Access Method

1. What are the data access requirements for standard file, message, and data management?
2. What are the access requirements for decision support data?
3. What are the data storage and the application logic locations?
4. What query language is being used?

6.5.5 Security Checklist

1. **Security Awareness:** Have you ensured that the corporate security policies and guidelines to which you are designing are the latest versions? Have you read them? Are you aware of all relevant computing security compliance and risk acceptance processes? (Interviewer should list all relevant policies and guidelines.)

2. **Identification/Authentication:** Diagram the process flow of how a user is identified to the application and how the application authenticates that the user is who they claim to be. Provide supporting documentation to the diagram explaining the flow from the user interface to the application/database server(s) and back to the user. Are you compliant with corporate policies on accounts, passwords, etc.?
3. **Authorization:** Provide a process flow from beginning to end showing how a user requests access to the application, indicating the associated security controls and separation of duties. This should include how the request is approved by the appropriate data owner, how the user is placed into the appropriate access-level classification profile, how the user ID, password, and access is created and provided to the user. Also include how the user is informed of their responsibilities associated with using the application, given a copy of the access agreement, how to change password, who to call for help, etc.
4. **Access Controls:** Document how the user IDs, passwords, and access profiles are added, changed, removed, and documented. The documentation should include who is responsible for these processes.
5. **Sensitive Information Protection:** Provide documentation that identifies sensitive data requiring additional protection. Identify the data owners responsible for this data and the process to be used to protect storage, transmission, printing, and distribution of this data. Include how the password file/field is protected. How will users be prevented from viewing someone else's sensitive information? Are there agreements with outside parties (partners, suppliers, contractors, etc.) concerning the safeguarding of information? If so, what are the obligations?
6. **Audit Trails and Audit Logs:** Identify and document group accounts required by the users or application support, including operating system group accounts. Identify and document individual accounts and/or roles that have superuser type privileges, what these privileges are, who has access to these accounts, how access to these accounts is controlled, tracked, and logged, and how password change and distribution are handled, including operating system accounts. Also identify audit logs, who can read the audit logs, who can modify the audit logs, who can delete the audit logs, and how the audit logs are protected and stored. Is the user ID obscured in the audit trails?
7. **External Access Considerations:** Will the application be used internally only? If not, are you compliant with corporate external access requirements?

6.5.6 System Management Checklist

1. What is the frequency of software changes that must be distributed?
2. What tools are used for software distribution?
3. Are multiple software and/or data versions allowed in production?
4. What is the user data backup frequency and expected restore time?
5. How are user accounts created and managed?
6. What is the system license management strategy?
7. What general system administration tools are required?

8. What specific application administration tools are required?
9. What specific service administration tools are required?
10. How are service calls received and dispatched?
11. Describe how the system is uninstalled.
12. Describe the process or tools available for checking that the system is properly installed.
13. Describe tools or instrumentation that are available that monitor the health and performance of the system.
14. Describe the tools or process in place that can be used to determine where the system has been installed.
15. Describe what form of audit logs are in place to capture system history, particularly after a mishap.
16. Describe the capabilities of the system to dispatch its own error messages to service personnel.

6.5.7 System Engineering/Overall Architecture Checklists

6.5.7.1 General

1. What other applications and/or systems require integration with yours?
2. Describe the integration level and strategy with each.
3. How geographically distributed is the user base?
4. What is the strategic importance of this system to other user communities inside or outside the enterprise?
5. What computing resources are needed to provide system service to users inside the enterprise? Outside the enterprise and using enterprise computing assets? Outside the enterprise and using their own assets?
6. How can users outside the native delivery environment access your applications and data?
7. What is the life expectancy of this application?
8. Describe the design that accommodates changes in the user base, stored data, and delivery system technology.
9. What is the size of the user base and their expected performance level?
10. What performance and stress test techniques do you use?
11. What is the overall organization of the software and data components?
12. What is the overall service and system configuration?
13. How are software and data configured and mapped to the service and system configuration?
14. What proprietary technology (hardware and software) is needed for this system?

15. Describe how each and every version of the software can be reproduced and re-deployed over time.
16. Describe the current user base and how that base is expected to change over the next three to five years.
17. Describe the current geographic distribution of the user base and how that base is expected to change over the next three to five years.
18. Describe how many current or future users need to use the application in a mobile capacity or who need to work off-line.
19. Describe what the application generally does, the major components of the application, and the major data flows.
20. Describe the instrumentation included in the application that allows for the health and performance of the application to be monitored.
21. Describe the business justification for the system.
22. Describe the rationale for picking the system development language over other options in terms of initial development cost *versus* long-term maintenance cost.
23. Describe the systems analysis process that was used to come up with the system architecture and product selection phase of the system architecture.
24. Who besides the original customer might have a use for or benefit from using this system?
25. What percentage of the users use the system in browse mode *versus* update mode?
26. What is the typical length of requests that are transactional?
27. Do you need guaranteed data delivery or update, or does the system tolerate failure?
28. What are the up-time requirements of the system?
29. Describe where the system architecture adheres or does not adhere to standards.
30. Describe the project planning and analysis approach used on the project.

6.5.7.2 Processors/Servers/Clients

1. Describe the client/server Application Architecture.
2. Annotate the pictorial to illustrate where application functionality is executed.

6.5.7.3 Client

1. Are functions other than presentation performed on the user device?
2. Describe the data and process help facility being provided.
3. Describe the screen-to-screen navigation technique.
4. Describe how the user navigates between this and other applications.
5. How is this and other applications launched from the user device?

6. Are there any inter-application data and process sharing capabilities? If so, describe what is being shared and by what technique/technology.
7. Describe data volumes being transferred to the client.
8. What are the additional requirements for local data storage to support the application?
9. What are the additional requirements for local software storage/memory to support the application?
10. Are there any known hardware/software conflicts or capacity limitations caused by other application requirements or situations which would affect the application users?
11. Describe how the look-and-feel of your presentation layer compares to the look-and-feel of the other existing applications.
12. Describe to what extent the client needs to support asynchronous and/or synchronous communication.
13. Describe how the presentation layer of the system is separated from other computational or data transfer layers of the system.

6.5.7.4 Application Server

1. Can/do the presentation layer and application layers run on separate processors?
2. Can/do the application layer and data access layer run on separate processors?
3. Can this application be placed on an application server independent of all other applications? If not, explain the dependencies.
4. Can additional parallel application servers be easily added? If so, what is the load balancing mechanism?
5. Has the resource demand generated by the application been measured and what is the value? If so, has the capacity of the planned server been confirmed at the application and aggregate levels?

6.5.7.5 Data Server

1. Are there other applications which must share the data server? If so, identify them and describe the data and data access requirements.
2. Has the resource demand generated by the application been measured and what is the value? If so, has the capacity of the planned server been confirmed at the application and aggregate levels?

6.5.7.6 COTS (where applicable)

1. Is the vendor substantial and stable?
2. Will the enterprise receive source code upon demise of the vendor?
3. Is this software configured for the enterprise's usage?

4. Is there any peculiar A&D data or processes that would impede the use of this software?
 - Is this software currently available?
5. Has it been used/demonstrated for volume/availability/service-level requirements similar to those of the enterprise?
 - Describe the past financial and market share history of the vendor.

6.5.8 System Engineering/Methods & Tools Checklist

1. Do metrics exist for the current way of doing business?
2. Has the system owner created evaluation criteria that will be used to guide the project? Describe how the evaluation criteria will be used.
3. Has research of existing architectures been done to leverage existing work? Describe the method used to discover and understand. Will the architectures be integrated? If so, explain the method that will be used.
4. Describe the methods that will be used on the project:
 - For defining business strategies
 - For defining areas in need of improvement
 - For defining baseline and target business processes
 - For defining transition processes
 - For managing the project
 - For team communication
 - For knowledge management, change management, and configuration management
 - For software development
 - For referencing standards and statements of direction
 - For quality assurance of deliverables
 - For design reviews and deliverable acceptance
 - For capturing metrics
5. Are the methods documented and distributed to each team member?
6. To what extent are team members familiar with these methods?
7. What processes are in place to ensure compliance with the methods?
8. Describe the infrastructure that is in place to support the use of the methods through the end of the project and anticipated releases.
 - How is consultation and trouble-shooting provided?
 - How is training co-ordinated?
 - How are changes and enhancements incorporated and cascaded?

- How are lessons learned captured and communicated?
- 9. What tools are being used on the project? (Specify versions and platforms). To what extent are team members familiar with these tools?
- 10. Describe the infrastructure that is in place to support the use of the tools through the end of the project and anticipated releases.
 - How is consultation and trouble-shooting provided?
 - How is training co-ordinated?
 - How are changes and enhancements incorporated and cascaded?
 - How are lessons learned captured and communicated?
- 11. Describe how the project will promote the re-use of its deliverables and deliverable content.
- 12. Will the architecture designs "live" after the project has been implemented? Describe the method that will be used to incorporate changes back into the architecture designs.
- 13. Were the current processes defined?
- 14. Were issues documented, rated, and associated to current processes? If not, how do you know you are fixing something that is broken?
- 15. Were existing/planned process improvement activities identified and associated to current processes? If not, how do you know this activity is not in conflict with or redundant to other Statements of Work?
- 16. Do you have current metrics? Do you have forecasted metrics? If not, how do you know you are improving something?
- 17. What processes will you put in place to gather, evaluate, and report metrics?
- 18. What impacts will the new design have on existing business processes, organizations, and information systems? Have they been documented and shared with the owners?

6.6 Architecture Compliance Review Guidelines

6.6.1 Tailoring the Checklists

- Focus on:
 - High risk areas
 - Expected (and emergent) differentiators
- For each question in the checklist, understand:
 - The question itself
 - The principle behind it
 - What to look for in the responses
- Ask subject experts for their views

- Fix the checklist questions for your use
- Bear in mind the need for feedback to the Architecture Board

6.6.2 Conducting Architecture Compliance Reviews

- Understand clearly the objectives of those soliciting the review; and stay on track and deliver what was asked for. For example, they typically want to know what is right or wrong with the system being architected; not what is right or wrong with the development methodology used, their own management structure, etc. It is easy to get off-track and discuss subjects that are interesting and perhaps worthwhile, but not what was solicited. If you can shed light and insight on technical approaches, but the discussion is not necessary for the review, volunteer to provide it after the review.
- If it becomes obvious during the discussion that there are other issues that need to be addressed, which are outside the scope of the requested review, bring it up with the meeting chair afterwards. A plan for addressing the issues can then be developed in accordance with their degree of seriousness.
- Stay "scientific". Rather than: "We like to see large databases hosted on *ABC* rather than *XYZ*.", say things like: "The downtime associated with *XYZ* database environments is much greater than on *ABC* database environments. Therefore we don't recommend hosting type *M* and *N* systems in an *XYZ* environment."
- Ask "open" questions; i.e., questions that do not presume a particular answer.
- There are often "hidden agendas" or controversial issues among those soliciting a review, which you probably won't know up-front. A depersonalized approach to the discussions may help bridge the gaps of opinion rather than exacerbate them.
- Treat those being interviewed with respect. They may not have built the system "the way it should be", but they probably did the best they could under the circumstances in which they were placed.
- Help the exercise become a learning experience for you and the presenters.
- Reviews should include detailed assessment activities against the architectures and should ensure that the results are stored in the Enterprise Continuum.

Index

ADM	27
Architecture Board	21
Architecture Capability	3
Architecture Compliance	31
checklists	39
guidelines	51
review	33
review process	36
Architecture Contract	27
Architecture Governance	9, 12, 21
implementation	18
Architecture Governance Framework	9, 13
structure	13
COBIT	12
COBIT framework	24
corporate governance	9
COTS	33
dispensation	15
governance	
characteristics	10
ISACF	12
IT governance	9, 11
IT Governance Institute	12
OLA	15, 26
operational-level agreement	15, 26
ORB	44
service-level agreement	15
SLA	15, 22, 30
Statement of Architecture Work	29
technology governance	9, 11

