World-Class EA:
A Practitioners' Approach to Developing Enterprise Architecture Following the TOGAF® ADM

A White Paper by:

Dave Hornford, Conexiam
Nathan Hornford, Conexiam
Sriram Sabesan, Conexiam
Sadie Scotch, Conexiam
Ken Street, Conexiam
Samantha Toder, Conexiam

February 2017

This White Paper is an informational document and does not form part of the TOGAF documentation set. Readers should note that this document has not been approved through the formal Open Group Standards Process and does not represent the formal consensus of The Open Group Architecture Forum.

Table of Contents

Table of Tables

Table of Figures

*Boundaryless Information Flow™
achieved through global interoperability
in a secure, reliable, and timely manner*

Executive Summary

This White Paper is a companion to the TOGAF® framework and is intended to bring the concepts and generic constructs in the TOGAF framework to life. This paper puts forward current thinking on developing, maintaining, and using an Enterprise Architecture (EA) using the TOGAF Architecture Development Method (ADM). It describes an approach based upon the established best practice contained within TOGAF, an Open Group standard.

The TOGAF framework is set apart from every other EA framework because it contains three central parts: a Method, a Content Framework, and an EA Capability framework. By design, it is also scalable and configurable. Configure the same concepts for the purpose and the scale of the EA Capability. The greatest strength of the TOGAF standard is that it provides a framework. This can be troubling for new Practitioners, who want a cookbook. The TOGAF standard does not provide a cookbook; it provides the essential scaffolding that different EA teams use to build their EA Capability.

This White Paper is structured to provide the context, content, and rationale behind choices and steps in the ADM, and supporting concepts in the TOGAF standard, that an EA Practitioner can consult at any point in time to develop, maintain, or improve the value extracted from their organization's EA. In short, this paper is intended to guide the Practitioner to use TOGAF essential scaffolding to deliver an actionable EA for their Enterprise.

The intended audience for this White Paper is as follows:

- Professionals who have been tasked with developing and evolving an EA

- Business leaders who are seeking to obtain value from an EA

- Enthusiasts in the field of EA or organizational transformation

This White Paper presents an end-to-end approach using the TOGAF standard in an Enterprise and serving four different purposes. As an end-to-end discussion, this paper does not address the detailed activity appropriate to a purpose and specific situation. As described in the TOGAF standard and in the rest of this White Paper, each phase provides guidance for answering a specific question. Some phases work together to reach a better answer to a question or problem statement. To understand the detailed set of activities in each phase, please refer to the TOGAF standard.

An EA that was developed for the purpose intended optimizes Boundaryless Information Flow™ within and between Enterprises based on open standards and global interoperability.

Part 1: Introduction

Introduction

Overview

This White Paper provides crisp, concise, and consistent guidance on using the TOGAF framework to develop, maintain, and use an Enterprise Architecture (EA). This paper is a companion to the TOGAF framework and is intended to bring the concepts and generic constructs in the TOGAF framework to life. This White Paper puts forward current thinking to develop, maintain, and use an EA that aligns to a set of requirements and expectations of the stakeholders and enables predictable value creation.

It is intended to take the TOGAF concepts and illuminate how each Practitioner can use the same concept to (a) deliver useful EA for their Enterprise and (b) deliver improvements to EA Capability. This point is important: *use the same concept*. Not the same technique, not the same template, not the same process. The same concept. For example, evidence from prevalent practice shows that there is not a single EA team that didn't use a repository, whether the repository is a file folder or a fully-fledged installation of modeling and analytic software. If you are struggling with this point, stop and think about the baggage you are carrying into the conversation. For example, while reading, if you have a reaction similar to "but a *real repository* includes …", ask yourself if this is universally true or are you carrying baggage into the conversation? The concept of a repository is universal; the implementation varies.

The essential scaffolding of the TOGAF framework is the concepts. Everything else in the TOGAF framework is either an example or a starter set to get you moving. Don't like the example, take advantage of the modular structure of the TOGAF framework and substitute it. Leading Practitioners and users do. This paper is about guiding the Practitioner in making the universal structure of the TOGAF framework work.

This White Paper is written for the Practitioner, the person who is tasked to develop, maintain, and use an EA. Choice of the term Practitioner is very deliberate, reflecting the role, rather than one of the myriad titles in an Enterprise the Practitioner may have.

This paper is structured to provide the context, content, and rationale behind choices and steps that an EA Practitioner can consult at any point. When effectively used, a thoughtfully developed EA optimizes Boundaryless Information Flow™ within and between Enterprises based on open standards and global interoperability.

This paper is explicitly about developing, maintaining, and, most importantly, using an EA. The range of potential Enterprises and purposes require a paper of this length to define the direction.[1] Following the approach suggested in the World-Class Enterprise Architecture White Paper (see References), the TOGAF standard is routinely applied to develop architectures supporting strategy development, portfolio management, project planning and execution, and solution development. Collective experiences reflect that

---

[1] See the definition of Enterprise in Definitions (on page 12). The important concept to keep in mind is that the term "Enterprise" is used as a boundary of analysis.

there is *no one right* EA deliverable, model, view, work product, or technique. Rather, the correct approach is specific to the purpose of the architecture development initiative. Anyone who suggests there is a single correct approach, model, view, work product, or technique is not providing the right advice for you to succeed. This paper will help you, the Practitioner, to identify the approach that is appropriate to any particular purpose.

Developing, maintaining, and using an EA requires deep interaction with several specialized functions such as strategy development, budgeting, benefits realization, portfolio management, program & project management, and operational units. This White Paper will:

- Introduce key topics of concern

- Describe the TOGAF standard concepts related to the topic

- Show how it is related to developing, maintaining, and using an EA

- Discuss what the Practitioner needs to know

- Describe what the Practitioner should do with this knowledge

Even though this paper has a logical structure, it is not simple task list. The depth and detail of the steps needed to be taken by the Practitioner are specific to the purpose and are iterative. The only variable is time spent for every step. As with all change work, listing what you need to know is not the same as defining the level of detail in the documentation.

Key decisions are made in an Enterprise following a business cycle. An architecture should inform and enable decision-making. Just align the delivery of architecture to the Enterprise's business cycle and the purpose of the architecture development initiative. The value is delivered when the architecture is used. It is plain and simple.

This White Paper is divided into six parts, as follows:

Part 1: Introduction

This part contains this introductory section and a set of definitions.

Part 2: Guidance on Enterprise Architecture

This part addresses:

- What an Enterprise Architecture is and what it is used for

- Coordinating EA development across the EA Landscape

- Coordinating EA development with the business cycle

Part 3: Guidance on Developing an Enterprise Architecture

This part addresses:

- Using the ADM

- Developing an Enterprise Architecture to Support Strategy

- Developing an Enterprise Architecture to Support Portfolio

- Developing an Enterprise Architecture to Support Project

- Developing an Enterprise Architecture to Support Solution Delivery

- Special Cases

Part 4: Guidance on Using an Enterprise Architecture

This part addresses:

- What to do when you are hip-deep in solution delivery

- Architecture in action (agile Enterprise, response to incident, etc.)

Part 5: Guidance on Maintaining an Enterprise Architecture

This part addresses:

- Managing multiple simultaneous roadmaps

- What to do when you are hip-deep in solution delivery

Part 6: Appendices

This part presents:

- A list of useful tables related to frameworks, reference models, etc.

How to Use this White Paper with the TOGAF Standard

The TOGAF framework provides essential universal scaffolding useful in a range of organizations, industries, and architectural styles. This paper is developed to fill in what is not explicitly addressed by the TOGAF framework and provides an approach to interpret the standard. This does not suggest that the TOGAF framework is flawed. The TOGAF framework is designed to require interpretation or customization. It has to provide universal scaffolding. What is common and universal between all of the different examples provided in the definition of Enterprise? Essential scaffolding expressed as concepts.

One way to look at the TOGAF framework is that it is written for the expert theoretician – the person who thinks about the structure and practice of EA. The TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References) is for the person tasked with establishing or evolving an EA Capability.

This paper is written directly for the person who does the work: develops, maintains, and uses an EA. The person who is not worried about the theory, and who is not worried about how to structure or maintain an EA Capability. The person who develops, uses, and maintains a good EA.

While this paper assumes no detailed knowledge of the TOGAF framework, it explores the core concepts of the TOGAF 9.1 standard. It places these concepts together in the context of using them to develop, maintain, and use an EA. This includes guidance on iteration, an EA Repository, executing the ADM for the purpose of supporting Strategy, Portfolio, Project, and Solution Delivery, and performing effective governance of the development and use of the EA practice.

We suggest learning the TOGAF framework's conceptual structures in the context of using them. We believe that the TOGAF framework is a useful tool, not a discrete educational topic. This paper follows the approach of exploring the conceptual structures in the context of making use of them. This paper assumes that you have established an EA Capability and have customized the TOGAF framework for your Enterprise.[2]

For a complete interpretation of the TOGAF standard, it is best to read this White Paper in conjunction with the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability. Together with this paper, there is a complete exploration using the TOGAF framework to establish an EA Capability and how that established EA Capability develops, maintains, and uses an EA.

Referenced Techniques

The techniques and key literature created by thought leaders that are actively used in leading practices are used to develop this White Paper. This paper limits references to materials freely available through standards organizations and academic publications. This paper does not promote or reference any commercial techniques, even that from the author's practice. There is often commercial material available for topics discussed in this paper. It is up to the reader to seek them.

References to key literature and their techniques are intended only to be representative. The reader is expected to read and assimilate referenced publications for a full understanding of these related topics. Some example work products are drawn from one of the best implementations using EA tools currently available in the market. This paper used these images of work products to highlight why it is used and what outcome is expected. Further, this paper does not suggest that the referenced tools, techniques, and literature are definitive. Other tools, techniques, and literature can readily be substituted. In fact, the referenced material is part of a body of knowledge that continuously evolves, and the reader is advised to explore updates to literature and techniques referenced in this paper.

See the Appendices for a set of sample starting points that can accelerate development of the EA practice.

---

[2] For assistance customizing the TOGAF framework, see the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References), which provides in-depth commentary and guidance for executing the Preliminary Phase of the TOGAF ADM.

Definitions

To share a clear understanding with the authors, a few terms need to be defined distinctly from common English usage. The terms below are distinctly defined, and capitalized wherever found. The authors mean exactly these definitions and nothing else in this document.

Enterprise

The highest level of description of an organization used to identify the boundary encompassed by the EA and EA Capability.

Note: This definition is deliberately flexible and not associated with an organization's legal or functional boundaries. It must cover monolithic organizations and extended organizations that include separate organizations connected by a mission or supply chain, as well as operating entities within an organization. Consider an organization that uses outsourced partners to provide manufacturing, logistics, and support; a multi-national peacekeeping force; and a multi-billion-dollar division of a Fortune 50 firm. All are Enterprises.

Enterprise Architecture (EA)

As the focus of this paper is to explain the TOGAF framework and the concept of Enterprise Architecture, it is better to define this concept in some detail. Succinct definitions tend to require specialized knowledge to understand the nuance. See The Purpose of Enterprise Architecture (on page 18) for a discussion of EA.

The two best definitions that can be used are from Gartner and DoDAF. Gartner[3] defines Enterprise Architecture as: "the process of translating business vision and strategy into effective Enterprise change by creating, communicating, and improving the key principles and models that describe the Enterprise's future state and enable its evolution". DODAF defines architecture as: "a set of abstractions and models that simplify and communicate complex structures, processes, rules, and constraints to improve understanding, implementation, forecasting, and resourcing".

While many in the EA profession find distinguishing the terms "architecture" and "architecture description" useful, this document does not make any such distinction. It is a distinction without a difference for a Practitioner.

Practitioner

The person tasked to develop, maintain, and use an Enterprise Architecture.

Note: This term reflects the role, rather than one of the myriad titles that may apply.

---

[3] See GartnerClarifies.pdf, ID: G00156559, August 12, 2008.

Part 2: Guidance on Enterprise Architecture

The Purpose of Enterprise Architecture

A quick perusal of the literature will rapidly highlight that there is no consistent understanding of what an Enterprise Architecture (EA) looks like, or how one uses an EA. Attempts to succinctly define EA speak of fundamental concepts, elements, relationships, and properties of a system. These attempts tend to carry a high level of specialized knowledge and often make little sense to non-specialists. Further, it can be argued that this is the result of many commentators focusing on the architecture they develop, with the implicit assumption that everyone should do the same. Understanding comes from purpose.

EA is a strategic tool that presents an approach to identify and address gaps between aspirations and reality, whatever drives the gaps. It accelerates the ability of an Enterprise to achieve its stated objectives. The tool comes with its method to use, taxonomy to support the directions, and resources needed to benefit from using the tool.

This section will address the following questions:

- Why is it important to develop an Enterprise Architecture?

- What is an Enterprise Architecture?

- How to use an Enterprise Architecture?

Why is it Important to Develop an Enterprise Architecture?

An EA is developed for one very simple reason: to guide effective change.

All Enterprises are seeking to improve. Regardless of whether it is a public, private, or social Enterprise, there is a need for deliberate, effective change to improve. Improvement can be shareholder value or agility for a private Enterprise, mandate-based value proposition or efficiency for a public Enterprise, or simply an improvement of mission for a social Enterprise.

Guidance on effective change will take place during the activity to realize the approved EA. During implementation,[4] EA is used by the stakeholders to govern change. The first part of governance is to direct change activity – align the change with the optimal path to realizing the expected value. The second part of governance is to control the change activity – ensuring the change stays on the optimal path.

The scope of the improvement drives everything that is done. A methodology that serves to validate both the objective and the change, ensuring that both are feasible, delivers the desired value, and in a cost-effective manner. An architected approach provides a rigorous planning and change governance methodology.

---

4 A common trap is getting into efforts to fix terminology by using a different synonym. This is always done when people have added meaning, or special conditions, to a word. Implementation means *"the process of putting a decision or plan into effect"*. Feel free to substitute transformation, change, program execution, or deployment if these words align with your preferences.

In its simplest terms, EA must describe the future state and the current state of the Enterprise. The description of the future state enables the right people to understand what must be done to meet the Enterprise's goals, objective, mission, and vision in the context within which the Enterprise operates. The gap between the Enterprise's current state and future state highlights what must change. A good EA facilitates effective governance, management, risk management, and exploitation opportunities. A list of gaps makes obvious what must change and the implications of that change: is the proposed project in alignment with what is needed? In alignment with priority? In alignment with the complete set of goals and objectives?

The preceding paragraphs highlight the conceptual scope of EA. This scope often leads to the assumption that EA is only used to answer the big questions. Nothing can be further from the truth. The same concepts, methods, techniques, and frameworks can readily be used to address the end state, preference trade-off, and value realization for big and little questions. The essential difference is not what you do; it is what the documented architecture looks like. The scope of the system varies; the detailed description of elements and properties vary. All of the concepts remain the same.

What is an Enterprise Architecture?

In short, EA provides the most effective path to realizing an Enterprise's strategy.[5] A good EA uses a holistic approach to translate strategy into a well-defined execution path, using appropriate analysis, planning, design, and implementation methods.

The purpose of EA is to enable the Enterprise to most effectively achieve the mission, business strategy, and goals through cycles of planning, design, deployment, and delivery of change. An architected approach provides a rigorous planning methodology that validates the business objectives, ensuring that they are feasible, deliver the desired business value, and their achievement is cost-effective.

Achieving this purpose comes from *understanding* the Enterprise, the context, the scope of change, and the value that will be realized. Using EA facilitates understanding. The Enterprise is described in consistent terms, highlighting fundamental parts and how they interact. Consistent terms enable like-with-like comparison. Potential changes to the fundamental parts are explored regarding the desired end-state and preferences. This understanding and analysis enable trade-off between competing preferences and potential changes that carry different costs and different benefits.

In short, a good EA enables stakeholders to knowingly strike the right balance between any competing set of preferences. It allows individual business units to innovate safely in their pursuit of business value delivery. At the same time, it ensures the needs of the organization for an integrated strategy are met, permitting the closest possible synergy across the extended Enterprise.

---

[5] Refer to Hambrick & Fredrickson: Are you Sure you have a Strategy? and Mintzberg et al: Strategy Bites Back (see References) for a very good discussion of what a strategy is. For the purposes of this paper, Hambrick's position is found to be best suited. He focuses on what a strategy is used for and defines it as the central integrated, externally-oriented concept of how an Enterprise will achieve its objectives. A definition that architecture can support.

Introduction to the EA Landscape

The TOGAF framework uses a concept of the EA Landscape to refer to the complete set of descriptions or the EA. This White Paper distinguishes EA Landscape from EA, because there will not be a single description in a comprehensive EA Landscape. At any point in time, a typical Enterprise will have several architectures described. Some architectures will address very specific needs; others will be more general. Some will address detail; some will provide a big picture. Some will address the same topics in different states (current, target, and transition), or different periods of time. To address this complexity, the TOGAF standard provides a framework for organizing the EA Landscape. The EA Landscape identifies the boundary of all potential architecture, and associated constraints and guidance.

Many characteristics can be used to organize an EA Landscape. An essential concept to recognize is that any initiative to develop and maintain EA populates part of the EA Landscape. Over time, over multiple actions, the EA Landscape is filled and refreshed. Much of the commentary on iteration in the TOGAF framework is designed to address this point. Unfortunately, the explanation is placed in the context of the ADM rather than the EA Landscape. By placing a linear time structure onto the concept of the ADM creates more problems than are solved.

Instead of considering iteration regarding re-sequencing and looping the ADM, combine the TOGAF concept of an Architecture Project with the concept of the EA Landscape. Every Architecture Project knowingly develops just enough of the EA Landscape to serve the need at hand. The development is done in the context of prior architecture that guides or constrains the current work. Each Architecture Project will create, refine, and potentially change components in the EA Landscape.

When populated, the EA Landscape contains a description, constraints, or guidance that can be used. Without performing repeated information gathering, analysis, review, and approval, the Practitioner cannot proceed with confidence. Existing decisions, guidance, and constraints inform current architecture development. Best practice limits information gathering and analysis to the minimum necessary to address the question at hand. Effort spent on EA returns the highest value when the EA is used. The EA cannot be used until the architect is "done". All architecture development must be assessed against Time-To-Market (TTM). Filling in only the required parts of the EA Landscape, and following the constraints and guidance already in place, speeds TTM.

Four common independent characteristics frame the EA Landscape:

- **Breadth**: The subject matter covered by an Architecture Project. Breadth is easy to find confusing since it can refer to a wide range of subjects. Consider domain, organization, and initiative as examples. Breadth can be a hierarchy of specific subject areas. For example, an organization can be broken down through the organizational hierarchy. Subjects are supple. For example, addressing a specific initiative will include all impacted organizations, and an organization will address all impacted initiatives. Breadth is one of the most important scoping dimensions. It provides the Practitioner the context of their analysis.

- **Level of Detail**: The level of detail should be self-explanatory. It is easy to get carried away to explore and elaborate continually within the scope of a domain, organization, or initiative. As the architectures are developed, elaborate to the extent needed to answer the question at hand. A good enough answer to support a decision or directionally guide is sufficient to make progress. Always develop to the least detail required to address the purpose of the Architecture Project. Always keep in mind that working on more detailed architecture is guided and constrained by less detailed or superior architecture.

Lastly, the more detail required, the longer the TTM. Detail takes time to gather, analyze, describe, and get approved.
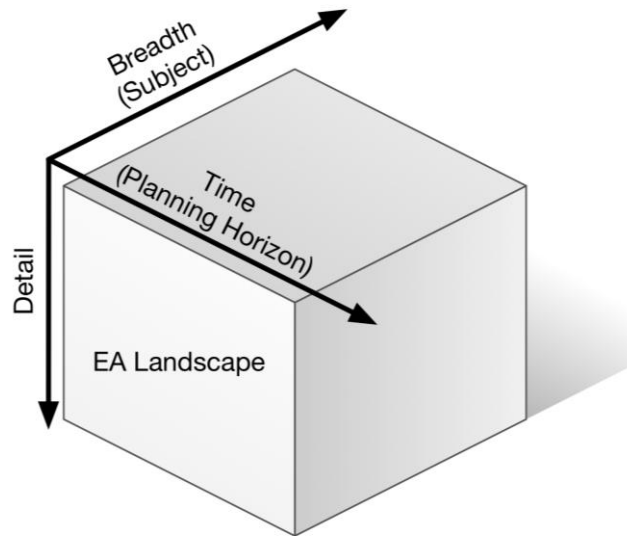
Figure 1: Characteristics of the EA Landscape

- **Time**: Every architecture development project will have a planning horizon; the point in time when you expect to reach the Target Architecture. Time creates challenges because the future is in motion. Typically, the longer the planning horizon, the less detailed the architecture. This is often true but does not provide a universal rule. Lastly, care must be taken where one or more transition architectures exist before reaching the planning horizon. The more detailed architecture must carefully conform to the guidance and constraints active at the point in time. This can be a challenge as the guidance and constraints change through different transition states.

- **Recency**: Each architecture description, specification, and view were created at a point in time. They are always built for a purpose, with an eye to the minimum information gathering and analysis to address the question at hand. All EAs age, often gracefully or suddenly. Recency is a hint that prior EA may need to be reviewed and either reaffirmed or replaced. A good repository can distinguish between architecture that is under development, architecture that has been approved, architecture that has been realized, and architecture that has been reaffirmed. During the development period, the architecture may be very current, but may not be valid for governance. Without approval, it should be considered as hints only. Recency may be used as an organizing factor for historical architectures.

The essential point is to recognize that EA Landscape contents are only developed when needed. Once approved, it constrains all further EA development and use of the EA. For a broader discussion of time, sequence, and business cycle, see Operating in the Context of Superior Architecture (on page 62).[6]

The dimensions of the EA Landscape help us think about the EA. Keep in mind that, in most cases, it is easy to build a simplification that is not valid. Architecture Projects are not neat cubes similar to what is shown in Figure 2. A real representation would look more like a sea urchin – a consolidated center but with spikes going in all directions.
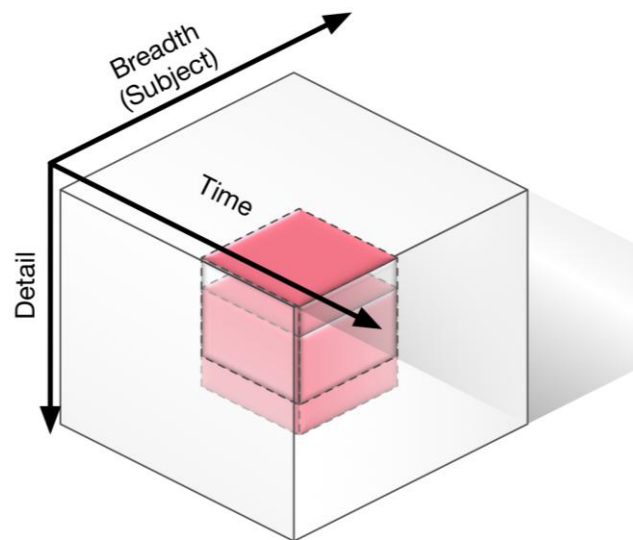
Figure 2: EA Landscape with an Architecture Project

Looking at Figure 2, the essential point is that the Architecture Project covers a specific portion of the EA Landscape – the portion defined regarding breadth, planning horizon, and detail. Prior work may already exist within the scope. The example does not cover the least or the most detailed layers, nor all time periods nor subjects. Rather the example addresses a specific portion of the landscape. The example Architecture Project will populate, or refresh, a portion of the EA Landscape. Because there is higher-level work, all work in the Architecture Project will be subject to the superior architecture. The example stops at a level of detail so the Practitioner will need to constrain the level of detail. Lastly, the example is within the total planning horizon of the Enterprise and will be constrained by what can and must be done within the planning horizon.

Complicating our lives, the superior architecture may exist either as an unrealized target, unrealized transition, or a realized current state. It must always be kept in mind that where there is not an explicit change in superior architecture, the current state probably remains valid. Lastly, this Architecture Project is a subset

---

[6] The term "superior architecture" is used to refer the architecture created for broader scope and purpose. For the Architecture to Support Portfolio, the Architecture to Support Strategy is the superior architecture. When traversing transition states, the reaffirmed Target Architecture is the superior architecture.

of the potential breadth of the scope of the EA Landscape. TTM is a key feature of useful architecture; Practitioners must stick to the scope (breadth, time, detail) of what they have been asked. Work outside the scope may be interesting, potentially even needed in the future, but is not within the scope of this architecture initiative.

The energy and efficacy of an EA team is diluted when it tries to be in every conversation by trying to do too much. The construct of a TOGAF Request for Architecture Work as the entry to Phase A exists to bound the current Architecture Project. The Request for Architecture Work tells the EA team that, within the context of the existing EA Landscape, its Enterprise is looking for a Target Architecture addressing a specific set of subjects at a necessary level of detail that can be accomplished within a particular planning horizon. A substantive output of the Architecture Project is to populate, replace, or reaffirm the contents of the EA Landscape. When stakeholders accept the target, all further EA work, change planning, and change execution are governed by the approved architecture.

Introduction to Purpose

A purpose-based EA Capability model identifies four purposes that typically frame the planning horizon, depth and breadth of an Architecture Project, and the contents of the EA Repository. The purpose-based EA Capability model used in this White Paper was introduced in the World-Class Enterprise Architecture White Paper (see References) and refined in the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References).

Architecture to Support Strategy	Architecture to Support Portfolio	Architecture to Support Project	Architecture to Support Solution Delivery

Figure 3: Purposes of Enterprise Architecture

Typically, there are four broad purposes of an EA Capability:

- **EA to Support Strategy**: Deliver EA to provide an end-to-end Target Architecture, and develop roadmaps of change over a three to ten-year period. An architecture for this purpose will typically span many change programs or portfolios. In this context, architecture is used to identify change initiatives and supporting portfolio and programs. Set terms of reference, identify synergies, and govern the execution of strategy via portfolio and programs.

- **EA to Support Portfolio**: Deliver EA to support cross-functional, multi-phase, and multi-project change initiatives. An architecture for this purpose will typically span a single portfolio. In this context, architecture is used to identify projects, and set their terms of reference, align their approaches, identify synergies, and govern their execution of projects.

- **EA to Support Project**: Deliver EA to support the Enterprise's project delivery method. An architecture for this purpose will typically span a single project. In this context, the architecture is used to clarify the purpose and value of the project, identify requirements to address synergy and future dependency, assure compliance with architectural governance, and to support integration and alignment between projects.

- **EA to Support Solution Delivery**: Deliver EA that is used to support the solution deployment. An architecture for this purpose will typically be a single project or a significant part of it. In this context,

the architecture is used to define how the change will be designed and delivered, identify constraints, controls and architecture requirements to the design, and, finally, act as a governance framework for change.

Architecture for different purposes typically creates different contents in the EA Landscape with a different mix of characteristics. Table 1 summarizes the typical characteristics. Table 1 is developed to represent a scenario, where a strategist uses the same concepts, methods, techniques, and frameworks to develop EA to develop a roadmap that supports the direction of an Enterprise. The strategist's Architecture Project will drill down from strategy to creating a portfolio that realizes the future state by supporting solution delivery. This table presents how the strategist or the architecture practitioner's work addresses the four dimensions of the EA Landscape.

Table 1: Purpose and EA Landscape Characterization

Purpose	Breadth	Level of Detail	Time	Recency
Architecture to Support Strategy	No pattern. Some Strategy will have a broad impact while other Strategy will cover a narrow subject.	Not very detailed. May contain point constraints that are very detailed when the value is dependent upon tight control. Typically, more guidance than constraint.	Typically, looking ahead for a 3 to 10-year period when Target. Current Architecture to Support Strategy tends to have a short timeframe of validity.	Typically, the need to update and keeping current this architecture is highly variable.
Architecture to Support Portfolio	Will cover single subjects (the Portfolio).	Typically, not very detailed. May contain discrete constraints that are very detailed when the value is dependent upon tight control.	Typically, valid for 2 to 5-year period when Target. Current Architecture to Support Portfolio should be considered past its best-before date. A portfolio without a view to the future is pointless.	Typically, the need to update and keeping current this architecture is highly variable.

Purpose	Breadth	Level of Detail	Time	Recency
Architecture to Support Project	Narrow breadth, typically discrete Projects within a Portfolio.	Typically detailed. Will contain detailed constraints, that may not be fully supported by detailed architecture descriptions. Typically, more constraint than guidance is developed.	Typically, valid as a target for <2 years. Will have very long-lived timeframes as current (post realization).	Typically, will be retained in the EA Landscape for an extended period after transition from Target to Current.[7] In the absence of an Architecture Project, the architecture and associated constraints and guidance will continue indefinitely.
Architecture to Support Solution Delivery	Typically, very narrow breadth.	Most detailed EA. Will contain the most detailed constraint. Typically, only constraints will be developed, as guidance will be carried forward from superior architecture.	Typically, valid as a target for <2 years. Will have very long-lived timeframes as current (post realization).	Typically, will be retained in the EA Landscape for an extended period after transition from Target to Current. In the absence of an Architecture Project, the architecture and associated constraints and guidance will continue indefinitely.

What an Enterprise Architecture Looks Like

EA exists to guide and constrain change planning and work to perform the change. The scope of work embedded in a Request for Architecture Work should identify the applicable characteristics of the EA Landscape. Over time, through multiple Architecture Projects, the EA Landscape is populated. This still does not tell us what actually gets written down, nor exactly what is produced.

In short, a Practitioner will need to document three things:

1. Models, in the EA Landscape

2. Views derived from the EA Landscape

3. Other useful things

In short, the architecture is the set of models, the components, and their relationships that comprise the scope of the EA Landscape under consideration. These models consistently describe the current and Target Architecture. In a theoretical world, a single unified model is produced. Typically, a set of models is produced. These discrete models will either have a jury-rigged linkage or rely on the expertise of those using

---

[7] A well-run EA Landscape will maintain components, as well as associated guidance and constraints, through their lifecycle. A typical lifecycle is to be introduced as a candidate, approved through governance as target, then convert to current following an Implementation Project.

the models to leap between them. Models can vary in formality, some strictly conforming to a semantically constrained structure, while others are quite flexible.

The primary purpose of the models is to facilitate the architect to understand the system being examined. Understand how it works today, understand how it can be most effectively changed to reach the aspirations of the stakeholders, and understand the implications and impacts of the change.

A secondary purpose is re-use. It is simply inefficient to re-describe the Enterprise. The efficiency of consistency is balanced against the extra energy to describe more than is needed, and to train those who describe and read the descriptions on formal modeling. The size, geographic distribution, and purpose of the EA team will dramatically impact the level of consistency and formality required.[8] Formal models are substantially more re-usable than informal models. Formal models are substantially easier to extend across work teams. The penalty is that formal models require semantic precision. For example, regardless of the structure of an application in the real world, it must be represented in a model conforming to the formal definition. This representation is possible with a good model definition.

Architecture Projects may have unique aspects. Practitioners usually lose the ability to address Architecture Project-specific considerations in a standard representation. The reverse is also true; flexible definitions that directly support one analysis will not be shared nor communicated with others in the EA team. Often the unique aspects will not even be remembered by the author. Practitioners must trade off between re-use and optimal fit, and should ensure that they are optimizing for the entire EA team rather than personal preference.

Every model that is produced and maintained has a price in effort. When effort exceeds value, the price will be paid by hindering an Enterprise's ability to perform the effective change. Unnecessary models and analysis steal from guiding effective change. Every approach to modeling is designed to shed light on one or more aspects of the Enterprise. Typically, narrow, special-purpose models facilitate detailed analysis while broad models facilitate inclusive analysis. All approaches to modeling – formal/informal and broad/narrow – are trade-offs.

All EA Landscapes that support a broad range of purposes will be comprised of a set of models. This set could be contiguous or discrete, targeted for analysis or communication. A core unified model can provide a common bridge between discrete models. The more specific a model, the more important it is to an analysis. The more important a model to analysis, the more important is the need and clarity of linkage across models. Careful thought is needed to understand the long-term need for cross-linkage. Most analyses are performed repeatedly over a period of time for different purposes. Like informal models, jury-rigged or expertise-based linkage is a short-term answer that prohibits effective re-use.

Models are very useful for the architect. They form consistent representations of the parts of the world that must be understood and analyzed. Shorthand communication and consistent analysis reduce the TTM.[9] However, because models are partial representations of the whole, typically described with a limited language that requires experience to read, and often subject to constraints designed to show relationships,

---

[8] See "Managing your Enterprise Repository" in the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References).
9 "Oh that process, it is a P3M, don't worry about it."

models tend to be ineffective to communicate usefully. Consider a balance sheet; it is a great model to outline part of an organization's financial position. It requires skill to read and is silent on the success, margin, or lifecycle of new products. Do not rush to deliver the models sooner than necessary.

Models are poor general communication tools. Good models are carefully constrained to exactly tell part of a story. They will carefully control the components available and the available relationships. They will enforce some attributes. They carefully render a complex environment into something that represents the world in terms it can be understood, optimized, and compared. They tend to require specialist knowledge, and often carefully constrain common terms in a way casual consumers do not align with.[10]

The best communication comes down to views, and "other useful things". Views have a specialized role in communicating the architecture and are discussed in Communicating with Stakeholders (Concern and View). The phrase "other useful things" is purposefully open-ended. For example, it is normal to find that a high fraction of useful communication is highlighting the value of the target state, acknowledgment of the scope of anticipated change, or clarifying the date value is expected. Most of the effective communication about an architecture will be "other useful things".

How to Use an Enterprise Architecture?

An EA is developed for one very simple reason: to guide effective change. Practitioners use models to provide a consistent analysis of complex systems. Models provide efficient long-term representation that enables like-with-like comparison – comparison of what is, what was, and what might be. The comparison that facilitates trade-off between potential changes that carry different costs and different benefits. Models provide understanding to people who understand the language, structure, and limitations of a model.

Guiding effective change is driven by who is using the architecture. Three broad communities use the EA: stakeholders, decision-makers, and implementers. Each of these communities uses the architecture differently.

When starting to talk about communication, the problem of terminology is the first obstacle faced. "Stakeholder" is a useful term, and multiple frameworks and methods use the term. Be aware of when you are carrying implied meaning from one framework, or approach, to another. This paper follows ISO/IEC/IEEE 42010:2011 guidance on stakeholders which focuses the attention on those whose concerns are fundamental to the architecture, or architecturally significant.[11] Facilitating effective communication

---

[10] For example, the term "strategy" is widely used; specifically within the OMG's Business Motivation Model. A high fraction of people who use the BMM trip over the term strategy. It holds a subordinate element in the model and the definition does not immediately resonate with common English. The BMM strategy definition "*represents the essential Course of Action to achieve Ends – Goals in particular; it is accepted as the right approach to achieve its Goals, given the environmental constraints and risks*".

[11] The term "stakeholder" is one of the most baggage-laden terms. Everyone who has an interest is not useful. ISO/IEC/IEEE 42010:2011 guidance on stakeholders in the context of an architecture description is useful. Stakeholders are those whose "concerns are considered fundamental to the architecture, or architecturally significant".

Part of the problem is formal definitions having to be broad to ensure that they properly include all reasonably conceivable stakeholders. In this paper where a formal definition doesn't provide pragmatic guidance, it will move promptly to pragmatic guidance, and leave the argument on semantic purity to those whose day-job pays for semantic purity. A Practitioner's day-job pays for producing useful architecture.

requires us to make a distinction between other communities who are interested in the architecture. A stakeholder holds approval rights on the target and the implementation; an implementer requires guidance and constraint; and a decision-maker holds execution rights on change. Practitioners are advised to develop views that address a stakeholder's concerns. Success of an architecture rests on the clarity and focus of the views produced. Its sole purpose is to communicate that the Target Architecture best satisfies the complex set of requirements the Enterprise has. Practitioners are best served when they preserve the distinction between stakeholders with approval rights and those needing most recent data points to create appropriate views of the concerns addressed by the EA. Without clarity on distinct roles, Practitioners complicate governance of the EA and the change projects.

Communicating with Stakeholders (Concern and View)

This paper provides practical advice to a Practitioner on using the TOGAF framework. Stakeholders' concerns and views are one area where the theoretical constructs embedded in the TOGAF standard are correct, but not directly translatable to use. The TOGAF standard takes a formal modeling approach to understanding stakeholder, concern, and view; this has led many to interpret that all representations of architecture are views prepared for any conceivable interest. In strictly technical terms that approach is correct, just not helpful,[12] considering usefulness and TTM. This paper will continue to hammer home the point "do just enough to support key decisions at this moment". Getting more data and providing more detail may sound appealing. The only thing an architect does not have is time. Do the right things to the best level of detail to market the architecture, and make people use the architecture. If there is time, pursue creating the rest of the views and elaboration to your heart's content.

Further, stakeholders, views, and concerns are often explained in terms of a single architecture. Consider what an EA Landscape will actually contain: Multiple discrete architectures. Separated by purpose, detail, breadth, time, and recency. And then there is architecture states: current, transition(s), and target. An architect's first obligation is ensuring the architecture addresses the preferences of the Enterprise. When the Practitioner preserves the stakeholder's concern, the view to communicate with the stakeholder, and how the architecture will address their concern, something useful to govern against in addressing this obligation naturally emerges.

From a practical perspective, consider:

- **Stakeholder**: Someone who has approval rights in the Target Architecture being explored by the current Architecture Project, and subsequently has decision rights to the suitability of the implementation.

- **Concern**: A consistent set of subjects that capture the stakeholder's interests and act to consolidate requirements.

- **View**: A representation of the EA Landscape that addresses a set of stakeholder's concerns. Either

---

The TOGAF 9.1 standard definition is: "an individual, team, or organization (or classes thereof) *with interests in, or concerns relative to, the outcome of the architecture*; different stakeholders with different roles will have different concerns".

The Project Management Institute (PMI) definition is: "an individual, group, or organization, *who may affect, be affected by, or perceive itself to be affected* by a decision, activity, or outcome of a project".

The ISO/IEC/IEEE 42010:2011 definition is: "an individual, team, organization, or classes thereof, having an interest in an Enterprise or system".

[12] Tell the inhabitants of Whitehorse, Yukon Territory that they live in southern Canada. Technically correct, but not helpful to any conversation with someone who knows they live in the North.

describe how the architecture addresses the concerns or demonstrate how the associated requirements are met.

The TOGAF concept of an Architecture Project provides context for both the development of new architecture and the change to realize it. By practically constraining the use of stakeholders to those with approval rights Practitioners enable governance, and more importantly governance in context.

This White Paper constrains the concerns to a topic and addresses the stakeholder's power, interest, and requirements against this topic. This approach surfaces topic-based decision rights and provides the ability to perform a trade-off between competing requirements. The sections discussing a walk through the ADM for different purposes will expand on the use of concerns. Pragmatically, most requirements will cluster in six to nine topic areas that are derived from the Enterprise's strategy. In fact, most concerns are consistent from one Architecture Project to another – they cluster around the central challenges the Enterprise is trying to address, such as agility, efficiency, IT complexity, or customer journey.

A consistent set of core concerns aligned to Enterprise priority facilitates focus on priority. Every Architecture Project brings to the fore Enterprise priorities and is in a position to demonstrate how this initiative is addressing the priority. Further, Practitioners are in a position to confirm consistency of requirement within a concern, and by stakeholder. Confirming consistency, or the lack, enhances the Practitioner's ability to discern the set of preferences the Enterprise is chasing.

Table 2 provides an extension of the TOGAF Stakeholder Map including concern and requirement. Missing requirements within a concern can either be a gap in information gathering or a demonstration the stakeholder is saying "this does not matter". Knowing requirement or lack of preference in relationship to power and interest directly facilitates trade-off. The trade-off is performed within a concern and between the concerns.

Table 2: Sample Stakeholder Map

	Concern 1			Concern 2		
	Power	**Interest**	**Requirement**	**Power**	**Interest**	**Requirement**
Stakeholder 1	High	Low		Low	High	
Stakeholder 2	High	High		Low	Low	
Stakeholder N	Low	High		High	Low	

Views address a stakeholder's concern about a specific architecture. In a perfect world Practitioners are able to use a single model directly. This is a mythical happy place. It will never be possible for a key issue such as agility or cost.

A view simply addresses a stakeholder's concern about an architecture. Often it is a potential architecture, and the view serves to help the stakeholder's potential target and associated change. This allows a stakeholder to put things in context and have confidence about the target and the change.

When stakeholders understand the architecture, the change, and the trade-offs, implementation governance is possible. Fail, and expect continuing issues as point answers highlighting one potential benefit without any compensating trade-off emerge throughout the planning and execution cycle.

When establishing the EA Capability, it is likely common classes of stakeholder were identified. If this was done essential concerns were likely identified.[13] These concerns represent the questions that the EA Capability is expected to answer, and may be considered mandatory. Successful high-functioning EA teams will maintain a library of viewpoints (see Appendix E: Sample Viewpoint Library on page 132) designed to address the questions they are expected to have answers for. Each viewpoint should identify the concern, the stakeholder(s), how the view should be constructed, and the information required to address the question.

Viewpoints are specialized communication to stakeholders that explicitly address a concern. Keep in mind that any associated requirements may not be satisfied by the architecture. The view is not a demonstration that the stakeholder should be happy; rather it is a demonstration of how the architecture addresses the concern.

Communicating with Implementers (Gap, Specification, and Control)

Implementers are typically poorly served. It is common to see implementers handed with a set of diagrams that represent the architecture. From these diagrams the implementers are expected to figure out the gaps they should fill, the architecture specifications they must conform to, and the controls they must implement. Implementers are better served when they are explicitly provided context, gap, architecture specification, and control.

The TOGAF standard identifies a very useful concept for communication with anyone implementing the Architecture Contract. An Architecture Contract identifies the responsibility of the implementation team to the Target Architecture's stakeholders. The most critical items to an implementer are:

- **Implementation Project context**: where does the project fit within the roadmap, what value or value dependency will the project provide?

- **Scope**: what work packages and gaps is the Implementation Project responsible for, as well as what gaps associated with any architecture components associated with the project scope is the project not responsible for?

- **Conformance**: what is the set of specific architecture specifications and controls the Implementation Project will be assessed against?

A sample solution delivery notebook from a leading EA consulting organization is presented in Appendix F: Solution Delivery Notebook (on page 133). The notebook presents the role of an Architecture Contract.

The essential component is to fulfill the purpose of the TOGAF Architecture Contract: link the Implementation Project to the target in terms of context, work required, and conformance test. Most critically, stop setting the implementers up by expecting them to work out what is expected and how the project's design and implementation will be assessed.

---

[13] See Customization of Architecture Contents and Metamodel in the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References), and Appendix D: Stakeholder/Concern Matrix on page 34.

John Carver's policy governance approach[14] is one of the best for a Practitioner to follow. There are two imperative practices in Carver to follow. First, specifications should be exclusionary, highlighting what is prohibited, rather than mandating what is permitted. Second, specification compliance should be assessed through a reasonable interpretation test by a reasonable person.

Drafting specifications as exclusionary reduces the requirement for omniscience during architecture development and provides the maximum opportunity for creativity during implementation, whether the creativity comes from innovative thinking by the design team, new technology, new third-party services, or new processes. Understanding what is prohibited, assumes everything else is allowed. The key concept is if the architecture does not constrain a choice, or prohibit a choice, the choice is allowed.

Given that creativity is encouraged, Practitioners cannot expect that an implementation team can read minds and implement in the same way as envisioned. This forces the compliance assessment to be a test of reasonable interpretation. The best practice is always to link a specification to a requirement.[15] This allows the design, or implementation, to be assessed against a requirement/specification pair. The specification is in the context of what motivated the specification. Following this practice, every specification exists to deliver something, and the implementation can be value tested.

When Practitioners serve the implementation team well, the stakeholders are supported. Practitioners provide the big picture to guide projects implicitly to value production, and requirement/specification pairs to guide the projects explicitly to value. In both cases, the value being produced is directly traceable.

Communicating with Decision-Makers (Other Useful Things)

The last community who must be communicated to are decision-makers. Typically, decision-makers will have a strong overlap with stakeholders. This distinction is necessary to ensure that the stakeholder/concern/view construct is restricted to the approval of the target. The ability to have crisp governance of the target and approval is too important to blur the line and include other communications.

Like communicating with implementers, communication to decision-makers often falls into the category of "other useful things". An architecture roadmap or the strategic architecture are empherical in nature. They are supported by conversations around "motivation statements", walking through the scope of the change, foundational nature of some of the Implementation Projects, employment of appropriate compliance report for decision support, etc. Such conversations fall under "other useful things". It may not be possible to create appropriate models to support these communications.

Decision-maker communication will typically be aligned with:

- Timing

- Trade-off decisions

---

[14] Refer to John Carver: Reinventing your Board (see References).

[15] In the case of a control, it is always associated to the risk for the same reason.

- Status

- Budget

- Compliance

- Confidence

Communication about timing is typically drawn from either the Roadmap, the Implementation & Migration Plan, or from Phase G. Timing speaks to when can the decision-maker expects activity to start, change something, complete something, or start to obtain value.

Trade-off decisions between stakeholders need to be communicated to others in the Enterprise. They are usually not involved in the trade-off. Communication about trade-off decisions is typically educational, serving to explain the trade-off decision. Critical conversations on trade-off by prior architecture and superior architecture will be held during Phase F, G, and H, informing decision-makers.

Status conversations are about the Architecture Project. The most important status conversations are about closing on an Architecture Vision in Phase A, resolving complex trade-off in Phases B, C, and D, and value, effort, and dependency conclusions regarding the Roadmap's work packages in Phase E. The status of value realization conversations will occur in Phase H. Depending upon the status of value, further conversations about architecture change requests, or initiating a new Architecture Project may occur.

Decision-makers have a deep interest in the budget. During Phase F's planning exercises some of the most complex trade-off decisions are made. Conversations with stakeholders during architecture and roadmap development revolve around value, effort, and risk. In Phase F spend is brought to the fore. Further, during Phase G budget control and availability will impact all Implementation Projects.

Best practice has decisions on non-compliance being made by stakeholders. They need to approve the recommendation to enforce the target, grant relief, or change the architecture. Communication about compliance is very similar to trade-off conversations. Also, when relief is granted, further conversations about scheduling a roadmap or implementation plan update should also occur.

Some of the most critical conversations with decision-makers are about confidence. The confidence they should have in the Roadmap and Implementation & Migration Plan, completing the change, and realizing the value. All architecture is an approximation; no Practitioner can underestimate the importance of confidence.

Conclusion

In order to guide effective change, Practitioners have to understand complex systems and analyze the possible ways to improve the complex system against a set of usually contradictory preferences. In order to understand and analyze a complex system, good Practitioners will represent the system in a set of models. These models are the architecture – a description of the system in terms of components and their relationships. Over time, through multiple Architecture Projects, the EA Landscape is populated.

Using an architecture requires translation of the models to a form that is useful to non-specialists. Practitioners should not expect stakeholders, implementers, decision-makers, or anyone else to understand the models' specialized language, structure, and limitations.

Practitioners need to communicate with three broad communities: stakeholders, decision-makers, and implementers. Each of these communities uses the architecture differently.

Stakeholders are presented with views that address their concerns. This enables stakeholders to understand the architecture, engage in trade-off decisions, and finally approve the Target Architecture.

Implementers need to understand their project. First, where their project fits within the roadmap, and its role in producing value. Second, what work packages and gaps they are responsible for, as well as associated gaps they are not responsible for. Third, how conformance will be assessed.

Decision-makers' communication often falls into the category of "other useful things", where Practitioners communicate timing of change and value, prior decisions, status, budget, and confidence. All Practitioners need to keep in mind that informal communication, outside the scope of models, architectures, views, roadmaps, specification, or compliance recommendations, are the most important communication that will be undertaken.

An effectively communicated architecture is one that provides confidence. The importance of confidence cannot be underestimated. Confidence that the architecture and associated roadmap of change is the guidance the Enterprise should follow. With confidence, an Enterprise's leadership will use the EA to direct and govern effective change.

Business Cycle

All organizations have existing change processes. The EA team needs to be aligned with the organization's planning, budgeting, operational, and change processes.[16] The Practitioner must understand that a theoretically perfect world where the EA team is engaged in all change cannot be expected. In practice, the scope of the EA team will be limited to some purposes, or will only be engaged in some changes. The TOGAF standard says you need to configure the ADM to align to your business. This is commonly interpreted to fit the ADM as an end-to-end process as an appendage to existing business processes. Instead, the architecture development processes need to feed, and support, the existing change processes. This means the ADM is used to deliver work products useful to other processes, and just enough of the ADM is used to deliver to other Enterprise processes.

Budget Cycle

For most organizations, the budget cycle controls change in the organization. Pragmatically, the EA team will be aligned to the budget cycle. Figure 4 shows a timeline view, depicting an alignment of key decisions made during a business cycle and the purpose architectures. EA for Strategy, Portfolio, and Project need to be completed before key milestones for budget decisions are made. EA for Solution Delivery is a continuous operation around budget control. The key takeaway is architecture *before the* decision. If you are trapped trying to architect after the decision, see Architecture after Decision.

Figure 4 provides a simplified budget cycle for structuring what is universal.:

- Budget Planning identifies what is needed and what new initiatives will be started.

- Budget Preparation is typically a top-down and bottom-up activity. Guidance about expectations and initiatives will be provided from the top, and each department will develop a spending request.

- Budgets provided are the subject of further decision-making. Allocating budgeted funds is a key step in executing change. A good budget is a financial embodiment of the organization's priorities for the current budget cycle. Prior to allocation to an Implementation Project everything is just an idea.

- Budget control is ongoing financial and benefits realization of an Implementation Project.

---

16 See Process Model in the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References).
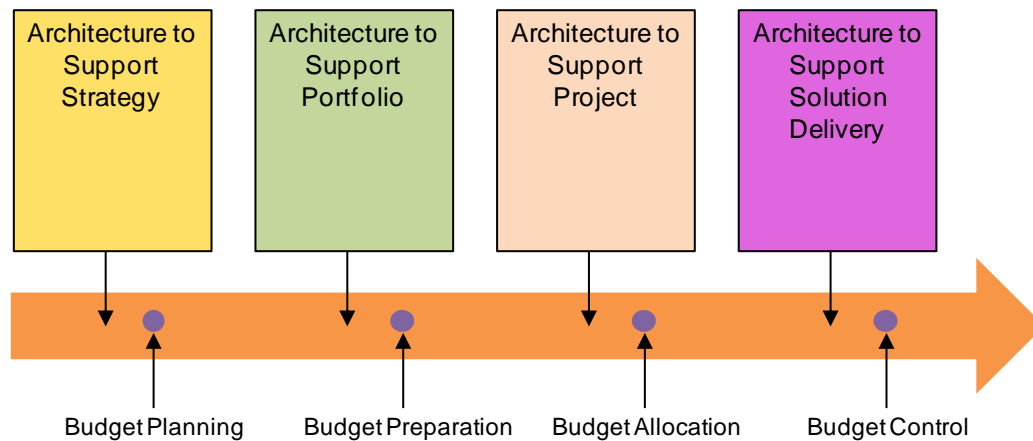
Figure 4: Business Cycle and Architecture by Purpose

Keep in mind that the simple unidirectional model allows us to see the interplay between key decision milestones. This paper uses the phrase "Architecture to Support" deliberately. The change process executes with or without a functioning EA team. The pragmatic question is what an EA team can do to guide effective change.

As mentioned earlier in this paper, it is best to tie everything to the budget cycle. The importance of good EA on guiding and constraining the change decisions is naturally noticed and highlighted. When there is no practical input from a good EA team before the decision an organization needs to take is made, the decision is still made. It might even be a good choice, but it was a less informed choice.

Keep in mind that in all EA the stakeholders, decision-makers, and implementers require effective support ahead of the decision. Good architecture that informs decision is infinitely more valuable than perfect architecture that follows decision and execution.

Budget Planning and Architecture to Support Strategy

The linkage between budget planning and Architecture to Support Strategy is a natural fit, that like many associations is not always correct. Part of the challenge is use of the term "strategy". Often the term is implicitly associated with the organization's strategy. Then without warning the same term is used for something far more specific, like the staff compensation strategy. At its most basic, a strategy is simply a "central integrated, externally-oriented concept of how to achieve the objectives".[17]

Like "stakeholder", a good definition encompasses a broad range of potential cases, without narrowing down to effective guidance. From an EA perspective, Practitioners are supporting strategy when exploring a longer-term target, and work will be used to identify a set of change initiatives. Guide the terms of reference for the initiatives so that the organization can direct and control execution through a portfolio of work.

---

[17] Refer to Hambrick & Fredrickson: Are you Sure you have a Strategy? (see References).

Typically, this type of work will align with budget planning, where the organization plans to spend on new initiatives or newly identified things. Table 1 (on page 24) identified that this work is typically only sufficiently detailed to provide guidance over a three to ten-year period and that the guidance will be valid for short periods of time. This is where organizations switch priority – the important element to recognize is the target is often rarely shifting; what is shifting is where priority is placed.

Good Practitioners know they are supporting strategy when the priority pendulum slows; when the organization is able to balance between two or more competing impulses. Effective guidance helps the organization understand what is required for the complete set of its needs.

Budget Preparation and Architecture to Support Portfolio

The linkage between budget preparation and Architecture to Support Portfolio is one of the strongest linkages available. Given a set of change objectives, the organization is embarking on what is a good approach – what work must be funded, what work can be deferred, and what work should be deferred. Some of the most powerful guidance to effective change an EA team can provide is to support portfolio planning and investment decision.

Providing Architecture to Support Portfolio requires working outside the corporate planning and execution cycle. When everyone else is executing on this year's budget, the EA team must be working on next year's budget; they have to be ready with a roadmap at the start of the budget preparation process.

The key questions every portfolio and budgeting process struggles with is a priority. Most portfolio and budget cycles are swamped in noise and cheerleading. They desperately need to know what work, in what areas must go forward and why. What work can be safely deferred? What work must proceed as a package? Some of the highest value work a Practitioner can provide is supporting portfolio and budget preparation.

However, it requires the roadmap to be available as the initial budget materials are being prepared, with an ongoing update from trade-off during the budget discussions. TOGAF Phase E and Phase F align directly to this use of Architecture to Support Portfolio. Phase E prepares the architecture roadmap for the budgeting process; work with all decision-makers in the budget preparation to finalize the Target Architecture, and the Implementation & Migration Plan.

A key use of the EA is to sustain a well-considered target. Budget and capacity to change determine what is planned for realization.

Budget Allocation and Architecture to Support Project

Architecture to Support Project is the first time you can see that work to effect change is about to be done. Before the release of funding to an Implementation Project, no change is going to happen. The classic alignment of this purpose in Phase F is the development of an Implementation Project business case or Implementation Project charter.

Architecture work facilitates the organization's final decision-making about the use of funding and other scarce change resources. The tendency of implementation teams to focus exclusively on the creation of tactical business value needs to be balanced with the roadmap purpose and value against the target. It is shockingly common for implementation teams to sacrifice substantive organization value to provide shiny baubles to the operational team the implementers work with.

Balancing the bottom-up change needs with broader initiative needs is an important role. Will the organization's priorities and values be realized by a particular Implementation Project? If so, the organization's budget allocation process should release the funds. If not, parochial departmental interests are capturing scarce organizational improvement resources. Frankly, the pattern of shiny bauble delivery is one of the most important reasons to perform Architecture to Support Project. If bottom-up business case justification built end-to-end efficiency, agility, or eliminated the need for transformation projects, no one would need the profession of EA.

The other role is ensuring completeness. Far too many projects build metaphorical half bridges; building everything but the last piece to cross the obstacle. The justification is usually to "make progress". Bluntly, an organization is not making progress when it embarks on a change it will not finish. The organization is simply wasting resources.

Figure 5: Half a Bridge

The TOGAF concept of the Architecture Contract provides the linkage between the value and the implementation through the target. The Architecture Contract provides traceability in terms of context, the complete work required, and conformance tests. See Appendix F: Solution Delivery Notebook (on page 133) for a sample of a Solution Delivery Notebook that fulfills the function of the Architecture Contract. Focusing attention on what will produce value and enabling architecture-supported governance is a chief outcome from Architecture to Support Project.

Budget Control and Architecture to Support Solution Delivery

Architecture to Support Solution Delivery is directly aligned with work to implement effective change.[18] In the business cycle, the budget control provides ongoing financial control and benefits realization. Architecture to Support Solution Delivery is directly aligned to the governance of the Implementation Project. Enabling direct association of spend with benefits realization is the contribution to the budget cycle.

---

[18] This paper is cognizant of repeated efforts to draw distinctions between "Enterprise Architecture" and "Solution Architecture", which seems to be driven by some attempts to associate EA to big thoughts and big initiatives. In practice it is a distinction that drives no changes in an effective EA team's organization and approach. This paper treats it as a distinction without a practical difference.

Architecture to Support Solution Delivery is dependent on traceability through the EA Landscape. Definition of acceptable boundaries for design and implementation, as well as boundaries for design and delivery, facilitate procurement and third-party contracting.

Similar to Architecture to Support Project, Architecture to Support Solution Delivery will use the TOGAF concept of an Architecture Contract to constrain design and implementation choices tightly to value.

Most Architecture to Support Solution Delivery will be performed in the TOGAF ADM Phase G. The need to fully iterate the ADM makes little sense when there is a superior architecture that develops the outline of the target, the stakeholders, a roadmap, and an implementation plan. If you are not getting value, you are creating busy-work and self-confusion about the ADM.

Business Cycle Conclusion

The business cycle is one of the core business activities that an EA team must align to. It provides a common reference point that is central to how an organization plans, authorizes, and executes change. Performing process alignment and alignment to other Enterprise frameworks is one of the central activities of establishing an EA Capability. For a broader discussion of other alignments, see the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References). This paper uses the business cycle as a simplification of the myriad of business activities that an EA team supports, to align with the practical work requirements of a Practitioner.

Coordination Across the EA Landscape and EA Team

This section will address the following questions:

- What to expect in a well-run Architecture Repository & EA Landscape

- How is ADM iteration realized in practice?

- How to work in the context of superior architecture

- How are multiple states managed (candidate, current, transition, and target)?

What to Expect in a Well-Run Architecture Repository & EA Landscape

First, a warning. In order to provide concrete examples of working in a repository, this paper presents few screenshots of one of the best implementations using a leading modeling tool. These represent one way that the challenges of a managing an EA Landscape in the real-world can be met. As outlined in Referenced Techniques (on page 15), this paper does not mean to suggest that the referenced tool, techniques, and literature are definitive. These concrete examples breathe life into the TOGAF concepts. You will have to extend the example to the problem you are solving. The Practitioner is advised to explore available practices and tools. This is a constantly evolving body of knowledge that improves and adjusts understanding.

The TOGAF standard identifies a broad set of materials that will be contained within the Architecture Repository. As a Practitioner, you will be directly concerned with the EA Landscape, Reference Library, Standards Information Base, Requirements Repository, and the Compliance Assessments in the Governance Log. Typically, these are implemented by a modeling and analytic tool, and a file repository.
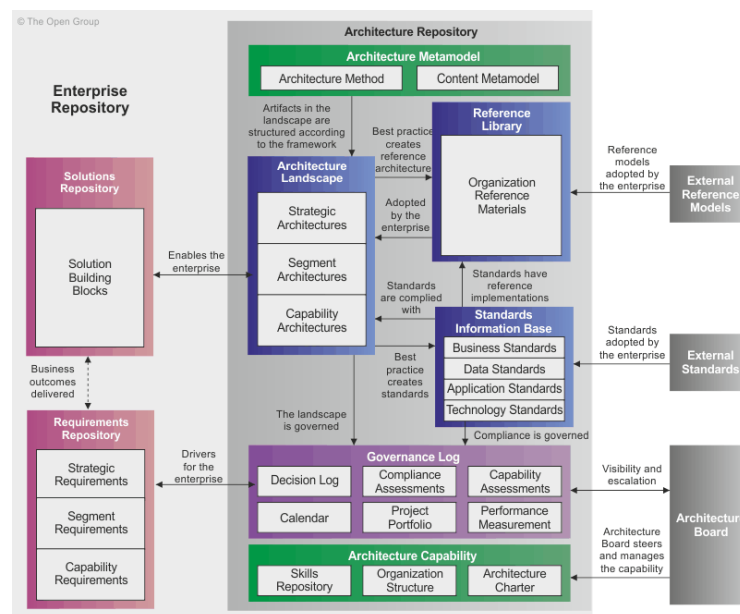
Figure 6: TOGAF Architecture Repository (Chapter 41)

A high-functioning EA team cannot deliver without using modeling and analytic software. Practitioners will continue to sketch diagrams casually as initial steps in understanding a system, or explaining one. Maintenance of a collection of sketches is not practical. It does not matter where they use a marker and 11" x 17" paper or spend hours connecting objects in drawing software, these sketches are not modeling and do not provide a meaningful contribution to the EA Landscape. Further, the gaps and errors inherent in casual sketching preclude considering the sketches as a model.

Do not confuse the guidance about managing an EA Landscape and EA Repository with commentary on effective communication. Most things an EA Capability needs to represent are complex. Visualization of complex situations to support the Practitioner, the stakeholder, and others that need to be communicated with is critical. Hand sketches are one of the most powerful communication tools available to a Practitioner. Beyond ideation is a serious error to present poorly thought-out visualizations to stakeholders and decision-makers. This paper strongly recommends the inclusion of information visualization skills in any EA team to address the needs of different communities – decision-makers, implementers, and stakeholders. One of the most significant challenges to developing a high-functioning EA team is overcoming poor information management and information presentation practice.

A significant factor that results in a well-run sustainable EA Repository is the ruthless minimization of information gathered and maintained. Any information that is not required for the current Architecture Project, or supports minimal traceability, should not be captured. EA teams routinely drown in an information overload after capturing and maintaining extraneous information – information that is typically only useful for more detailed architecture analysis or implementation. Good Practitioners will not confuse ruthless minimization of work with skipping necessary work: all stakeholders' concerns must be addressed. Leading Practitioners will understand that stakeholder management is necessary and attention to non-key stakeholders is rarely on the critical path.

The three most powerful components of an EA Repository are the Architecture Requirements Specification, controls, and gaps. Managing the transition from levels of detail can be greatly simplified when, instead of modeling for the sake of building a comprehensive end to end model, its integrity is preserved, avoiding incomplete analysis for areas of the architecture where sufficient detail is not available. When there is sufficient detail to guide and constrain, the Practitioner's work is done.

The test of sufficiency is a function of fitness for purpose. Best practice governance has the architect demonstrate that the views produced for the stakeholders and any constraints and guidance are derived from the architecture. Stakeholders approve views, not architecture descriptions.

More detail is always available to be captured and represented in the architecture model; additional model kinds; additional refinement. When a Practitioner models for the sake of modeling, there is no endpoint. The test of success is whether the stakeholder's concern can be addressed. As an example, the Enterprise is attempting to improve agility – can the view demonstrate to the satisfaction of the stakeholder that this Target Architecture and all associated change delivers agility? When sufficient information is gathered, and analyzed to demonstrate agility, the Practitioner is done. When the implementer can be provided with a list of gaps that need to be filled, Architecture Requirements Specifications, and controls that must be followed, the Practitioner is done. Do not do the work that comes after the decision, or activity that you are currently architecting to support.

A high-functioning EA team will be supported by modeling and analytic software, as well as a document management system. Whether these software functions are provided in a single suite or a set of software tools

is not material. A Practitioner requires the linkage between any models and documentation, as well as a space to perform necessary analysis to develop their candidate architecture.

What is produced is either a work product that is actively consumed or the intermediate work products the Practitioner needs to produce the requested work product. Table 3 provides a summary of work products that are actively consumed by key Enterprise processes.

Table 3: Partial List of Work Product Alignment with Key Processes

Practice Supports	Architecture to Support Strategy	Architecture to Support Portfolio	Architecture to Support Project	Architecture to Support Solution Delivery
Phase A Work Product: Vision	Key deliverable Before framing of a strategic planning session Refresh before initiation of program budgeting	Key deliverable Before start of budget planning	Often not used Activity to produce a vision overlaps with portfolio/program candidate architecture and roadmap Technique may be used at initiation of business case	Limited use Primary use is early in implementation cycle (via internal providers or execution partners)
Phase E Work Product: Candidate Architecture	During strategic planning session Refresh as required in program budgeting	Key deliverable Before start of budget planning Primary use is stakeholder acceptance of target and definition of gap	Before project initiation and finalization of business case Primary use is creation of Architecture Requirements Specification	Before engagement of execution partners (including internal providers) Primary use is creation of Architecture Requirements Specification
Roadmap	During strategic planning session Refresh as required in program budgeting	Before start of budget planning Refresh as required to support budgeting and program management	Limited use Can be used as an input to projects with multiple interactive changes	Before engagement of execution partners (including internal providers) Primary use is identification of required change, and preferences of how to execute change, to manage solution delivery partner selection and engagement
Phase F Work Product: Architecture Contract & Architecture Requirements Specification	Likely not used	Limited use	Key deliverable Before completion of project initiation	Key deliverable Before engagement and contracting

Practice Supports	Architecture to Support Strategy	Architecture to Support Portfolio	Architecture to Support Project	Architecture to Support Solution Delivery
Implementation & Migration Plan	Likely not used	During portfolio budgeting Refresh as required to support budgeting and program management	Key deliverable Before project start	Key deliverable Before engagement and contracting
Phase G Work Product: Conformance Assessment	Likely not used	Likely not used	Key deliverable At key points in project that allow reporting to stakeholders and obtaining decisions for non-conformance	Key deliverable At key points in project that allow reporting to stakeholders and obtaining decisions for non-conformance
Phase H Work Product: Value Assessment	Before governance review, framing a strategic planning session and program budget	Key deliverable Before governance review and program budgeting Refresh as required to support program management	Limited use Scope of significant architecture change and value often does not cleanly align to projects	Limited use Scope of significant architecture change and value often does not cleanly align to solution deployment

Successful Practitioners will strictly follow the first step of the architecture development phases (Phase B, Phase C, and Phase D) that says to select appropriate viewpoints. In order to select viewpoints, the Practitioner needs to know the stakeholder and concern. From these, the viewpoint that addresses the stakeholder/concern pair will identify the information necessary to address the stakeholder's concern. Any information that is not required information to address a stakeholder concern should not be gathered and analyzed. Extra information is pointless.[19]

When the Practitioner focuses on effective communication with stakeholders, implementers, and decision-makers, pointless activity is eliminated.

What to Expect in a Well-Run EA Repository: EA Landscape

One of the most challenging aspects of a well-run repository is managing transitions over time. In most simple terms, every architecture will exist in up to four states. The current state is what exists in the Enterprise today; this baseline provides the reference for all change. The target state[20] is what stakeholders

---

[19] At several points in this White Paper, and other papers from the same authors, there are very strong statements about effective architecture practice. These statements are drawn from the experience of the authors and reviewers. Gathering, maintaining, and analyzing pointless information is no different than establishing an EA team for the wrong purpose. Eventually, it will be fatal for the EA team.

[20] Earlier this paper used the term "end state". In reality, there is no end state for an Enterprise, unless it is terminating its operations. The paper also used "future state" to indicate lapse of time to achieve and experience the improvement. From this point onward this paper will use "target state" to

have approved; this state provides the reference for governing all change activity. Transition states are partially realized targets between the current state in the target state. The candidate state is what has been developed by the EA team but has not been approved for a status sufficient to govern change.

In practice, transition and candidate states create the most complexity in an EA Repository. Conceptually exploring gaps is easy; only look at what changed between the current and target states. Consider the four characteristics of the EA Landscape: breadth, depth, time, and recency. Now mix in multiple states. Now mix in that as time progresses the architecture can change. Now mix in that different Architecture Projects can work on the same subject at different times and different levels of detail. Variability is the nub of the information management problem. To be able to see the best set of required changes, the Practitioner must ruthlessly minimize the information maintained, and maximize the use of decision records.
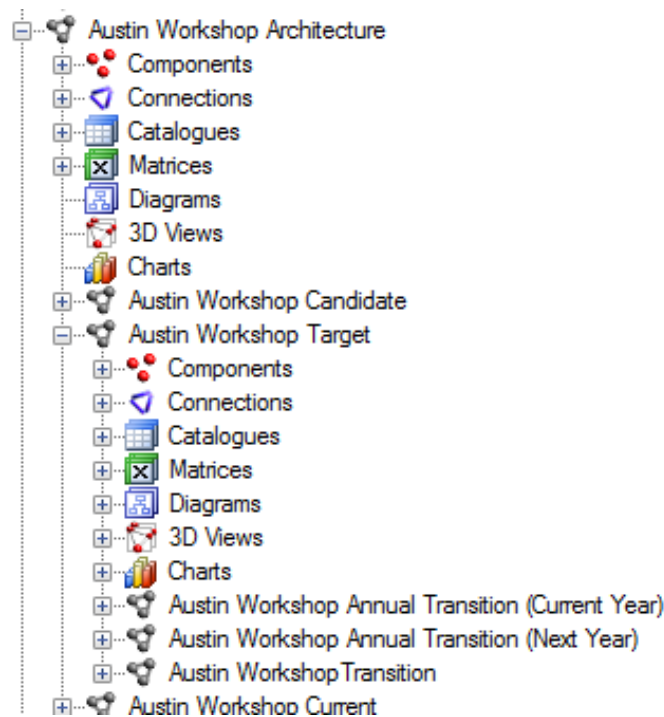
Figure 7: Example EA Repository

Figure 7 is a screenshot from a real-life EA Repository. A common current state description of the architecture is maintained in the repository. This common current state is periodically updated and used as the basis of all gap analysis. The governance test is that the current state reasonably represents what is. The repository also contains a consolidated target state and several transition states. When Architecture Projects come to a close, their architecture descriptions are moved into the consolidated target state. As the current state, the consolidated target is used in all gap analysis. While there is variance between transition states in the consolidated target, the Practitioner is in a position to assess whether the current project is moving towards the Enterprise's preference.

---

indite that it is the foreseeable best case scenario the Enterprise is striving to achieve. Having achieved, the same concepts and approach for trade-off can be applied or fine-tuned to new scenarios.

Architecture under development creates an additional information management challenge. For every Architecture Project, create a separate container in the EA Repository. This container allows the Practitioner freely to explore candidate target state options, different trade-off decisions, and impacts without affecting any other Practitioner's work. A well-run EA Landscape will perform its modeling and analysis to support the decisions/questions at hand only to the extent necessary and nothing more. These practitioners understand and execute with the notion that more detailed work would come from another architecture cycle, post-decision to discuss implementation. Figure 8 has separate architectures for an Architecture Project exploring a Portfolio, Project, and Solution Delivery.
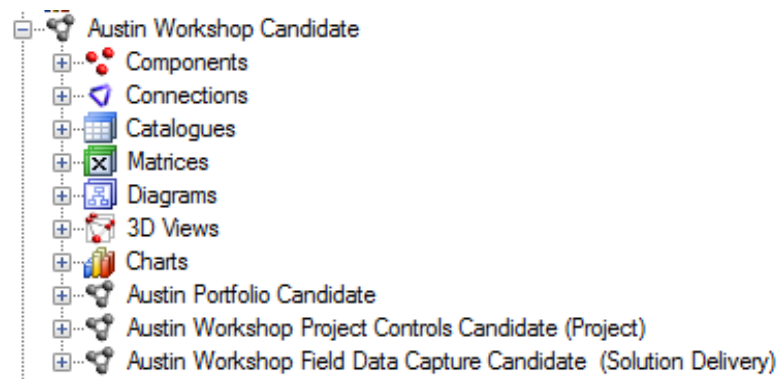
Figure 8: Multiple Candidate Architectures

Figure 7 and Figure 8 provide an example. Different EA modeling and analytic software, or even a different approach in an EA tool, would have different screenshots. The essential component is ensuring that the EA Repository supports different states, and provides flexibility for an architect to explore a potential future without impacting any other architect's work.

Supporting documents maintained must clearly identify their state. Without this ability, the Practitioner is pragmatically uncertain whether the document they are looking at is relevant, valid, or useful. They must readily allow the Practitioner to determine their recency. In practice, a candidate or target, or distantly realized current state architecture might be useful to the Practitioner. Usefulness is predicated on the "self-identification" of state and timeline. Without such markers, each supporting document is nothing but noise.

What to Expect in a Well-Run EA Repository: Reference 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 Reference Library of a well-run EA Repository is filled with accelerators. Accelerators speed time to market. A recurrent theme in this paper is ensuring sufficient architecture work is produced to support decisions and actions about the Enterprise's change activity. The most precious resource in change activity is time.

There is a broad set of reference materials used by a Practitioner. Broadly there will be two sets of reference material distinguished on whether they are directly used in architecture development, or provide background material. The first are materials that are used within the EA Landscape. These will include reference models, reference architectures, and patterns. These reference materials provide proven approaches. Proven approaches are accelerators, as they do not need to be explored with the same rigor as a novel approach. For

example, the IT4IT standard and APQC's Process Classification Frameworks.[21] In both cases there is no need to invent a novel set of processes. This type of reference material provides a complete starter set, simplifies communication, and enables re-use within the EA team. Each Practitioner will use the same terms to describe a problem. Figure 9 provides an example of reference material available in an EA Repository to improve architecture development.
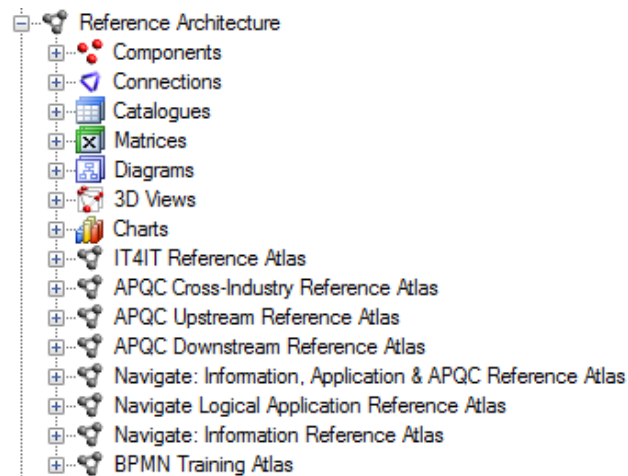
Figure 9: Reference Material in Modeling and Analytic Tool

Patterns, and other Architecture Building Blocks (ABBs), are typically indistinguishable to a Practitioner from other reference material in the EA Landscape. Whether brought in from reference sources, or created inside the organization, they provide a consistent and known way of approaching a problem.

The second set are documentary reference materials. This material may include white papers, discussions of EA Landscape reference material, templates, stock material, and guides. Again, reference material is an accelerator. Communication between Practitioners is improved when they have access to consistent background thinking. Communication outside the EA team is improved with consistency.

Figure 9 is a screenshot showing different reference architectures, and reference models, as discrete architectures. Maintaining discrete architectures allows the architect to be able to compare how the reference architecture was used in the current candidate or target against the base reference material. In longer-lived repositories, it is common to find multiple overlapping reference architectures. Consider an organization that uses APQC's Process Classification Framework as a base reference model. Should they implement a mainstream ERP, they will likely have work produced in the ERP vendor's process classification and the system integrator's process classification. Later, when the same organization adopts the IT4IT reference architecture, they will likely have another process classification.

---

[21] American Productivity and Quality Center; refer to: www.apqc.org.

Maintaining each of these has a clear reference in the modeling, and analytic software will allow future architects to understand the decisions made during architecture development and implementation governance, especially when only part of a reference is brought into architecture development and maintained in the architecture. This paper acknowledges the need to integrate an architecture tool with tools supporting planning, solution delivery, solution validation, etc. A Practitioner may have to refer to documentation in such tools on occasion or provide appropriate traceability. The family of tools and integration is beyond the scope of this White Paper.

Reference architectures, planning data, analytic data, etc. are normally supported by detailed documentation managed in a document management system. A Practitioner concerned with the purpose and rationale for complete or partial use of such data will seek the supporting documents, to use them appropriately for modeling or analysis. Do not get swayed by looking at whether the Practitioner is likely to read them when creating the links to the document management system.

What to Expect in a Well-Run EA Repository: Standards Information Base

In a well-run EA Repository, the standards information base will perform two functions. First, it provides a repository for the standards that the architecture must comply to. Second, it provides a repository for the standards imposed on all implementations by the architecture. The distinction is critical. One is used to test the architecture; the second is used to test an implementation.

In practice, these two sets of standards have to be separated. A simple example is provided by the PCI standards. An Enterprise that uses credit cards is subject to PCI standards. No Enterprise with a good EA will simply place PCI standards in a repository for an implementation to comply with. The question of how to comply is inappropriate for an implementation team. The compliance with PCI may be as simple as a standard derived from the EA that requires the use of a third-party payment processor ensuring that PCI subject information is not in the hands of the Enterprise. The latter is a standard derived from the EA.

It is common to extend the standards information base to include selected products and third-party services. This pragmatic choice simplifies the governance of Implementation Projects where, in addition to an architecture requirement specification or control, there exists a product or service that conforms. To further the example above, rather than the Architecture Requirements Specification requiring the use of the third-party payment processor, a specific third-party payment processor can be placed in the standards information base.

Where specific products and services are placed in the standards information base, it is best practice to trace those choices directly to the Architecture Requirements Specification or control that brought these products and services to life. Without traceability to the architecture, product or service selection can be viewed as an arbitrary choice. One of the traps of architecting through product and service standards is the lack of traceability to the requirement or risk. When there is simply the specification of a product or service as an arbitrary choice, the governance process is dramatically complicated because alternative products or services can be considered on criteria other than those that lead to an architecture supported decision.

What to Expect in a Well-Run EA Repository: Requirements Repository

Managing requirements to the entire EA Landscape is one of the most complex activities facing the Practitioner. The first challenge is simply the breadth of detail; the second challenge is the overlapping nature of managing requirements across the EA; the third challenge is maintaining the repository over time; and potentially the fourth is integrating with other repositories.

One thing that is important to consider is that requirements appear radically different depending upon the purpose of the architecture and the level of detail. As an extreme example, Practitioners with experience in solution delivery architecture and implementation may not recognize requirements for architecture developed to support strategy as requirements. Practitioners used to implementation tend to be looking for very granular requirements to express statements of need. Be agile, be efficient, integrate the new division, and protect the market-leading differentiators are all examples of key requirements for Architecture that supports Strategy and Portfolio.

Leading practices find that a large number of requirements for Architecture that supports Portfolio and Project are normally captured in the form of scores. Ask the stakeholders to assess the required efficiency, maturity, automation of a process, application, service or capability; score the required business fit or technical fit of applications; and score the preferred lifespan of the infrastructure. Best practice is to use a scale of one to five to capture their assessments. All of these scores are requirements; they clearly state the preferences of the stakeholders.

An important question in any requirements repository is whether these are architectural requirements or implementation requirements. The distinction can be fine, but it is a distinction with a very large difference. One of the tests that can be used for distinguishing between architecture and implementation design is whether the description can only be done one way, or can it be realized multiple ways. The former tends to be architecture, while the latter is implementation design. When an Architecture Repository is integrated with a requirements repository for implementation, use appropriate integration options to maintain traceability and integrity.

Many architecture requirements are remarkably long-lived. Especially when the requirement is articulating aspects of the Enterprise that differentiate it. When does a market leader who leads through customer experience want to relax the requirement requesting best-in-class customer experience? The real challenge for the Practitioner is translating market-leading customer experience into clear architecture specifications applied to components in the architecture. Herein lies one of the mental challenges when architecting for different purposes – the line between a requirement and a specification may be in who stated it. A requirement into a portfolio architecture aimed at market-leading customer experience may result in an architecture specification requiring that the information object "customer preference" be a common information object to the CRM, customer portal, and service desk. That specification reads like a requirement to the architect supporting solution delivery of the new CRM.

Requirements from higher in the organization also tend to be discussed using different names. It is common to speak of objectives and mandates, and treat them with special reverence. Likewise, the distinction between types of requirement – functional *versus* non-functional, business requirements *versus* technical requirements – is treated very seriously. In the final analysis, whether a requirement is a mandate, a non-functional requirement, or a business requirement, from the perspective of a Practitioner it is a statement of need that will be addressed in the context of the superior architecture and the set of objectives provided by all stakeholders.

One central activity Practitioners typically are not comfortable doing is assessing the validity of requirements. When the Practitioner has a well-described strategy, a portfolio that identifies gaps, and gap-filling work packages, it becomes easy to look at a requirement being injected in the project or solution delivery architecture and assess whether this requirement is in conformance with what the Enterprise priority is or whether this requirement conflicts with the superior architecture. Consider a portfolio initiative focused on improving agility for customer experience: this portfolio will identify a set of projects explicitly designed

to improve some aspect of the customer experience and improve the ability of the Enterprise to change. As time progresses close to execution, it is common for requirements not aligned with the project's purpose to be injected into the process. The central element of requirements management is good governance. Practitioners are guardians of the statements of value.

When Practitioners have a good architecture identifying the target and transition steps along the way, requirements, and architecture specifications, may vary over time; be different in the target and the transition architectures. Imagine a portfolio roadmap that deliberately sacrifices customer experience for agility in the first transition. Then in the second transition the priority switches and agility is scarified for customer experience. The conformance test to architecture requirement, and guidance on priority, switches. This paper deliberately uses the term "sacrifice" because inherent in this requirements repository is clarity of precedence and priority. When clarity of precedence and priority is not available, data to guide trade-off early in the cycle is absent, hindering progress. Just as the assessment of precedence and priority shifts context to other decisions where a set of preferences are well defined and is closer to the organization most suited to make the choice.

Explicitly link the architecture specification to requirements, and trace the requirements to a stakeholder/concern pair to track the value and preference. This traceability is used in governance to assess how well the design and implementation choices address the stakeholder's value preferences.

Best practice EA Repositories facilitate traceability at every step of the architecture to the direction and priorities of the Enterprise. Practitioners are delivering some of the highest value when they are engaged in requirements management and trade-off. All smart stakeholders want all, want more, and for free. All smart stakeholders know they can't have it all, nor can they have it for free. What stakeholders don't know – and what the role of the Practitioner is – is to assist the stakeholders in understanding what they have to give up in order to realize different sets of preferences.

A Practitioner with a well-run EA Repository is in a position to maintain a comprehensive set of requirements in context. Requirements in context enable the Practitioner to work actively for the preferences of the stakeholders rather than architecting to a subset of the preferences of the stakeholders; or worse a set of preferences that the Practitioner personally prefers.

What to Expect in a Well-Run EA Repository: Compliance Assessments

Most EA Repositories are missing the most important component of a compliance assessment: gaps, Architecture Requirements Specifications, controls, and views that address concerns stakeholders find interesting. A well-run EA Repository will contain all of the components necessary to perform effective compliance assessments as well as the compliance assessments.

The first step of compliance assessment is clarity on what compliance will be assessed against. Best practice compliance assessments are tightly linked with the TOGAF concept of an Architecture Contract. The Architecture Contract identifies what an Implementation Project is expected to deliver and the set of constraints the project operates under. Without clearly documented expectations and constraints the Practitioner has failed the implementation team.

A well-run EA Repository will contain the equivalent of an Architecture Contract for every Implementation Project. See Appendix F: Solution Delivery Notebook (on page 133) for an example of an Architecture Contract. With clarity on expectation and constraint, compliance may be assessed.

TOGAF Phase G identifies two areas where compliance is assessed. The first is the scope of the project. Second is the actual implementation, whether designed or the performance change. Phase H contains a further value-based compliance assessment.

The first assessment in Phase G considers the scope of the Implementation Project compared to the gap, or work package, expected to be filled. The work package identifies which gaps are going to be filled. The singular purpose of the work package is clarifying the work necessary to address the gaps in the architecture. Good roadmaps developed as part of an Architecture Project support portfolio will house well described work packages. Well described work packages are clear about gaps being filled, and the implementation strategy, or approach, of how the gap will be addressed. Where there is no architectural significance, no good Practitioner will bother constraining an Implementation Project with unnecessary guidance or constraint through the implementation strategy. Where the approach to addressing the gap is significant, a good Practitioner will always provide the appropriate guidance of constraint.

Performing scope, and implementation approach, compliance is the first step in protecting value. A good EA will provide clarity about the best path to maximized value for the Enterprise. Typically, maximized value to the Enterprise will not align with parochial preferences of the Implementation Project sponsor, or the implementation team. Frankly, if there was alignment, there would not be a need for an EA team. It follows that assessing the scope of an Implementation Project is the first place to protect value. Waiting until the project is funded and underway is indistinct from developing architecture after the decision; see Architecture Governance (on page 117).

The second Phase G compliance assessment confirms whether specific Architecture Requirements Specifications have been followed. The TOGAF concept of an Architecture Requirements Specification identifies what must be, what must be done, and what is prohibited. It provides the set of constraints on more detailed architecture development, design, and implementation.[22]

Phase H's compliance assessment is based on value realization. Typically, expected value will not be realized for a significant period of time after an Implementation Project has declared victory. Using the linkage provided by the Architecture Contract, recurrent value realization assessments can be performed. Maintaining the linkage from specification to stakeholder expectation facilitates consistent review.

Although a well-run EA Repository will be focused on demonstration of realizing value, traditionally most attention is placed on rule-following compliance. While rule-following is important, it tends to struggle with a consistent demonstration of value, unless it is assumed the value of following the rule is self-evident. Rule-

---

[22] An Architecture Requirements Specification can be delivered through different levels of detail and in multiple ways. For clarity, this paper distinguishes use of an architecture specification to address a stakeholder requirement, from a control to address a risk. The semantic distinction is used to assess for value. Typically, stakeholder requirements have an up-side, where risks have a downside. This paper typically divides architecture specification into four types:
- Principle: used to provide guidance on how to think about the decision.
- Pattern: used to provide a reusable approach to the decision.
- Standard: used to specify a correct approach to the problem.
- Rule: used to specify a correct answer and eliminate any decision.
The level of constraint required determines the type used by the Practitioner.

following compliance assessment is common where the Architecture Requirements Specification eliminates all design and implementation choice. Focusing assessment on rule-following is also most likely to be tied to requests for relief from the rule because the total cost of the rule is not in alignment with available value; see Architecture Governance (on page 117).

Best practice is to go beyond simple compliance with the statement, to include compliance with intent. The purpose is again to protect the expected value of the Target Architecture. When a constraint is connected to a stakeholder requirement, the compliance assessment is able to assess how well the design and implementation choices deliver on expected value. Compliance assessments that indicate the implementation will fail to enable expected value are key inputs to future architecture development.

How is ADM Iteration Realized in Practice?

An often-misunderstood element of the TOGAF framework is the ADM and the concept of iteration. The TOGAF ADM graphic provides a stylized representation that is often misinterpreted as a linear waterfall process model. This approach leads to some of the most confusing diagrams and explanations. The TOGAF ADM is a logical method that places key activity steps together for the purpose of understanding relationship of activity and clarifying information flow. The classic TOGAF crop-circle diagram is a stylized path that demonstrates essential information flow.

The TOGAF ADM should not be understood as a processes model. The ADM graphic is a stylized representation showing essential information flows and is not a representation of activity sequence.

The important thing to realize is every time the EA team is undertaking any activity within the scope of the ADM it is executing a Phase and developing the contents of the EA Landscape. For example, if a Practitioner is working on roadmap development, the Practitioner is exercising the steps in the TOGAF ADM Phase E (Opportunities and Solutions). The Practitioner needs to consume the mandatory inputs and produce the mandatory outputs. This applies to all ADM phases.

Start with recognizing that the inter-dependent nature of developing a Target Architecture requires considering the entire architecture, resulting gaps, and resulting work to clear the gap simultaneously. No Practitioner can consider a change, without considering the impact on all other domains, the resulting set of gaps, and the resulting set of work to clear the gap.

Unfortunately, describing that level of interaction is not practical. To address the complexity, the TOGAF framework provides an ADM phase for each essential output. Best practice ensures Practitioners use effective information inputs and produce useful outputs.

Depending on what a Practitioner is requested to develop, an architecture for the Practitioner's work plan will vary. Consider the impact on which phases of the ADM would be used for the following requests:

1. Given that the organizational design, customer interface, and processes are to be left unmodified, what other changes would allow "moving to the cloud"?

2. What changes are required to switch from more than 50 independent organizations pursuing small projects, to an integrated company capable of organizing, and controlling, construction projects 100 times larger than the current average?

3. What changes are required to the core claims platform to allow a 300% growth in customers and

transactions, and enable continuous change to policy terms?

4. Given that the ERP and current Finance & HR processes will be kept, what are the minimum changes to support allocating labor to capital projects?

5. How to integrate the acquisition with the minimum change, while sustaining both the current high-efficiency processes and the unique capability from the acquisition?

6. How to enable a third-party developer's agile approach, and Microservices, on the customer intimacy project?

7. How to modernize a particular platform without impacting anyone outside IT?

Each of these requests has been addressed using the TOGAF framework, and the techniques. Each started with a different purpose, and each traversed a distinct path that used a different configuration of the TOGAF ADM.

The only exception is Phase A; the Practitioner must start with Phase A. An Architecture Project must be initiated.

Phase A: The Starting Point

All architecture development needs to start with Phase A. Without the set-up inherent in Phase A Practitioners can expect to slide off-course and fail to deliver useful architecture.

The set-up essentials of Phase A are:

- **Define the scope of the Architecture Project**: What problem are you solving? In terms of the EA Landscape (breadth and planning-horizon) and in terms of purpose, which will tend to confirm the necessary level of detail? Be completely clear where in the business cycle this architecture will be used.

- **Identify stakeholders, concerns, and associated requirements**: Explore the EA Repository for superior architecture constraints and guidance. Do the Stakeholder Map. Be completely clear which stakeholders must be served and what they are worrying about.

- **Assess the capability of the EA team**: Take a hard look at the EA team and confirm the ability of the team to deliver on this architecture development project. A good EA team covers gaps in experience, skill, and bias to deliver the architecture that is useful, overcoming weaknesses of few members of the team.

The completion essentials of Phase A:

- **Key stakeholder agreement on a summary of the target and the work to reach the target**: Perform sufficient architecture development in all domains to enable you to communicate to the key stakeholders how the problem you have been assigned can be addressed and the scope of change to reach their articulated preferences. Be clear on the target, the value of the target, and the work to change.

Frankly, Phase A is routinely skipped, or skimmed. Good Practitioners know the key stakeholders agree on the summary target, the value, and the effort of change before any detailed work is undertaken. If key

stakeholders won't agree at the outset, they are unlikely to agree after the Practitioners have performed a lot of work detailing what they do not want, delivered insufficient value, or will not agree to change.

Completing the outputs of Phase A requires exploring all of the domains – whether the exploration is to understand what should change, or where change is not an option to determine the impact of retaining current architecture.

Practitioners should not be surprised if there are multiple potential targets after the initial exploration. Having more than one approach to addressing the problem is acceptable to key stakeholders. It facilitates better trade-off when performing more detailed analysis. Keep in mind that until the target is finalized, the Practitioner is exploring the best potential future, not selling a particular future.

Essential ADM Output and Knowledge

A summary of the essential outcome and output is provided in Table 4. Keep in mind that the essential output is what stakeholders, sponsor, and boss' boss' boss wants. No-one wants an architecture; they want guidance on planning and executing an effective change. Practitioners use an architected approach to providing the best available guidance on effective change. The essential outcomes and outputs are derived from the objectives of the phase – the statement of why a Practitioner should perform this activity.

What the Enterprise values and consumes is typically different than what the Practitioner produces. Practitioners deliver an essential output. It is provided as views, roadmaps, architecture specifications, controls, and other useful things. Architecture is developed, and the EA Landscape populated. To do this, Practitioners require a set of essential knowledge. The Enterprise consumes effective guidance about and the ability to govern change.

Read Table 4 in conjunction with Table 3 to confirm whether for a particular purpose the output of the phase is already in existence, needs to be created, or is extraneous to the current Architecture Project. Good Practitioners will adjust their work accordingly. Table 4 lists only key outputs and outcomes. For an exhaustive list, refer to the TOGAF standard. In order to achieve these outcomes, the Practitioner may have to perform more activities or create more deliverables than those listed in the table below. The intent is to keep the focus on what is pursued, not what is done.

Table 4: Essential ADM Outputs, Outcomes, and Required Knowledge

Phase	Output & Outcome	Essential Knowledge
Phase A: Architecture Vision	Sufficient documentation to get permission to proceed. Permission to proceed to develop a Target Architecture to prove out a summary target.	• The scope of the problem being addressed. • Those who have interests that are fundamental to the problem being addressed. (Stakeholders & Concerns) • What summary answer to the problem is acceptable to the stakeholders? (Architecture Vision) • Stakeholder priority and preference. • What value does the summary answer provide?

Phase	Output & Outcome	Essential Knowledge
Phase B, Phase C, & Phase D	A set of domain architectures approved by the stakeholders for the problem being addressed, with a set of gaps, and work to clear the gaps understood by the stakeholders.	• How does the current Enterprise fail to meet the preferences of the stakeholders? • What must change to enable the Enterprise to meet the preferences of the stakeholders? (Gaps) • What work is necessary to realize the changes, that is consistent with the additional value being created? (Work Package) • How stakeholder priority and preference adjust in response to value, effort, and risk of change. (Stakeholder Requirements)
Phase E: Opportunities & Solutions	A set of work packages that address the set of gaps, with an indication of value produced and effort required, and dependencies between the work packages to reach the adjusted target.	• Dependency between the set of changes. (Work Package & Gap dependency) • Value, effort, and risk associated with each change and work package. • How stakeholder priority and preference adjust in response to value, effort, and risk of change.
Phase F: Implementation and Migration Plan	An approved set of projects,[23] containing the objective and any necessary constraints, resources required, and start and finish dates.	• Resources available to undertake the change. • How stakeholder priority and preference adjust in response to value, effort, and risk of change. (Stakeholder Requirements)
Phase G: Implementation Governance	Completion of the projects to implement the changes necessary to reach the adjusted target state.	• Purpose and constraints on the implementation team. (Gap, Architecture Requirement Specification, Control) • How stakeholder priority and preference adjust in response to success, value, effort, and risk of change. (Stakeholder Requirements)

---

[23] Do not fixate on definition of the term "project" or what a project is. It is just an organizing effort for work to achieve an understood outcome. Your organization's internal definition of a project, and the label used, will be unlikely to align with anyone else's. My assistant refers to booking a flight as a project.

Phase	Output & Outcome	Essential Knowledge
Phase H: Architecture Change Management	Direction to proceed and start developing a Target Architecture that addresses perceived, real, or anticipated shortfalls in the Enterprise relative to stakeholder preferences.	• Gaps between approved target, or preference, and realization from prior work. (Value Realization) • Changes in preference or priority. (Stakeholder Requirements)

Iteration

The ADM provides a model of activity that supports producing the essential output by producing one or more work products. The central question determines whether there is a need for the essential purpose of a phase on a particular Architecture Project. If so, you will enter the phase at some point in time. If the essential purpose is not needed or has already been addressed, then this Architecture Project does not enter the phase.

Most commentary in the TOGAF standard on the iteration of the ADM is designed to address the point that if the Practitioner does not have the information at hand in the EA Landscape, the information must be produced. Unfortunately, these commentaries speak in terms of activity rather than output. Instead of considering iteration in terms of re-sequencing and looping the ADM, the Practitioner should explore the EA Landscape. If the information required, in terms of subject, detail, time, and recency is available – move on. If not, produce the material required. To produce material, the Practitioner is exercising a TOGAF ADM phase.

As an example, see the stylized Gantt chart in Figure 10. This figure provides a process-oriented view of executing the ADM. The Gantt shows the inter-dependent nature of EA requires all ADM phases that develop a candidate architecture and test it for acceptance to be open simultaneously. The ADM phases stay open to address the information required; once it is provided they close. Also, regardless of where the Practitioner is in time or purpose or Architecture Project, if the Business Architecture is being developed the Practitioner is executing Phase B. Executing Phase B is all about addressing the stakeholder concerns from the perspective of the Business Architecture domain, identifying the gaps in the Business Architecture, and looking at impacts across the EA Landscape. The figure highlights that many of the steps in the ADM phases can be executed simultaneously. Good Practitioners will explore impacts and address stakeholder concerns across the entire architecture.[24]

---

[24] This does not suggest that one person does it all. Developing an EA is a team sport with specialist positions. Following the analogy, the team has to play the same game at the same time.
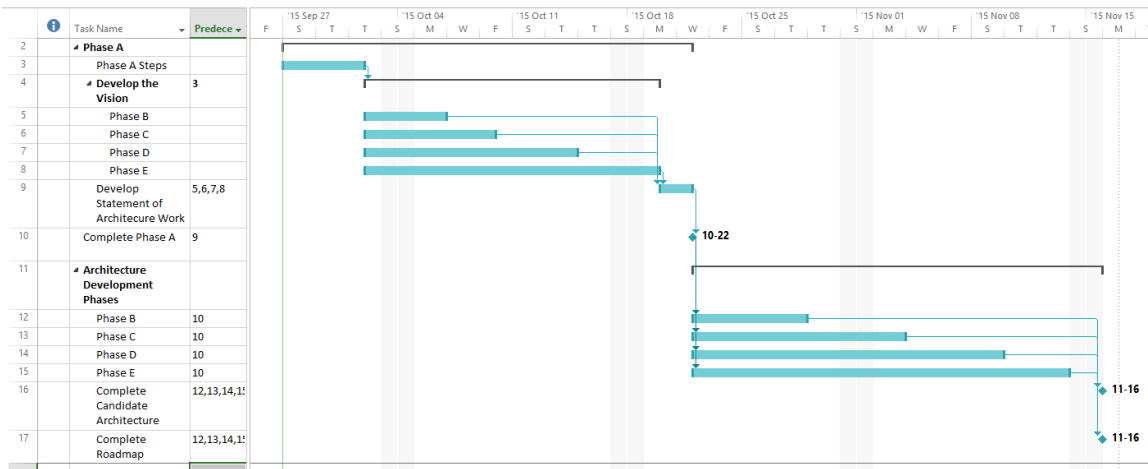
Figure 10: Stylized Architecture Development Gantt Chart

Consider the different purposes and a cascade through time as shown in Figure 4. When the plan in the stylized Gant chart in Figure 10 is applied to each purpose, it becomes clear that the Practitioner continually revisits the required phases, at the appropriate level of detail.

Most of the normal problem-solving models provide linear approaches with step gates. The linear approach helps us understand the process, and may represent the business cycle stage gates. However, they do not represent how people actually solve problems. Figure 11 is derived from Jeff Conklin's Wicked Problems & Social Complexity within Dialog Mapping (see References), and outlines a standard linear problem solving progression and how professionals typically address a problem. Testing the concept and potential implementation interactively is a best practice. Iteratively considering whether the high-level direction makes sense in terms of execution, and does execution make sense in terms of high-level direction?
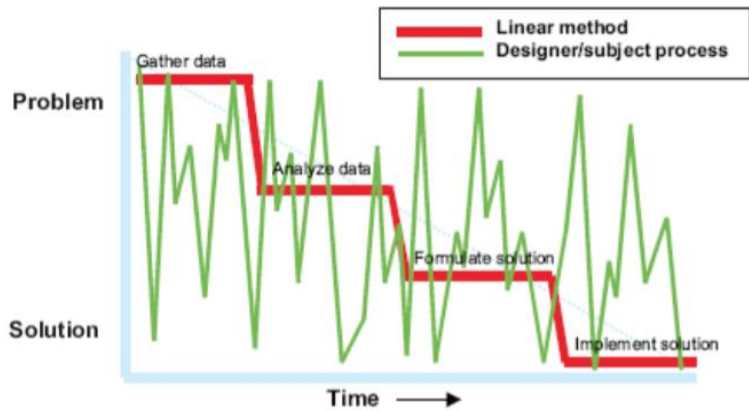
Figure 11: Problem Solving Approach (Derived from Conklin's "Wicked Problems")

All iteration is driven by the information needs of the current project. The process created is not dependent upon the work the EA Capability undertakes to produce, but the timing of completion. The essential question is when an EA Capability must deliver specific work products. Table 3 provides a summary of work products that are actively consumed by key Enterprise processes.

ADM Plan for Architecture to Support Strategy

The path to developing an Architecture to Support Strategy is a configured journey through the ADM. This path follows this journey:

- Understand context

- Perform assessment and analysis

- Define approach to target state

- Finalize Architecture Vision/target state

The processes iterate through the ADM to deliver an architecture that clarifies a Target Architecture roadmap of change over a three to ten-year period. The roadmap will identify change initiatives and support portfolio and programs. It will set terms of reference for the initiatives and identify synergies. A key use is governing the execution of strategy via portfolio and programs.
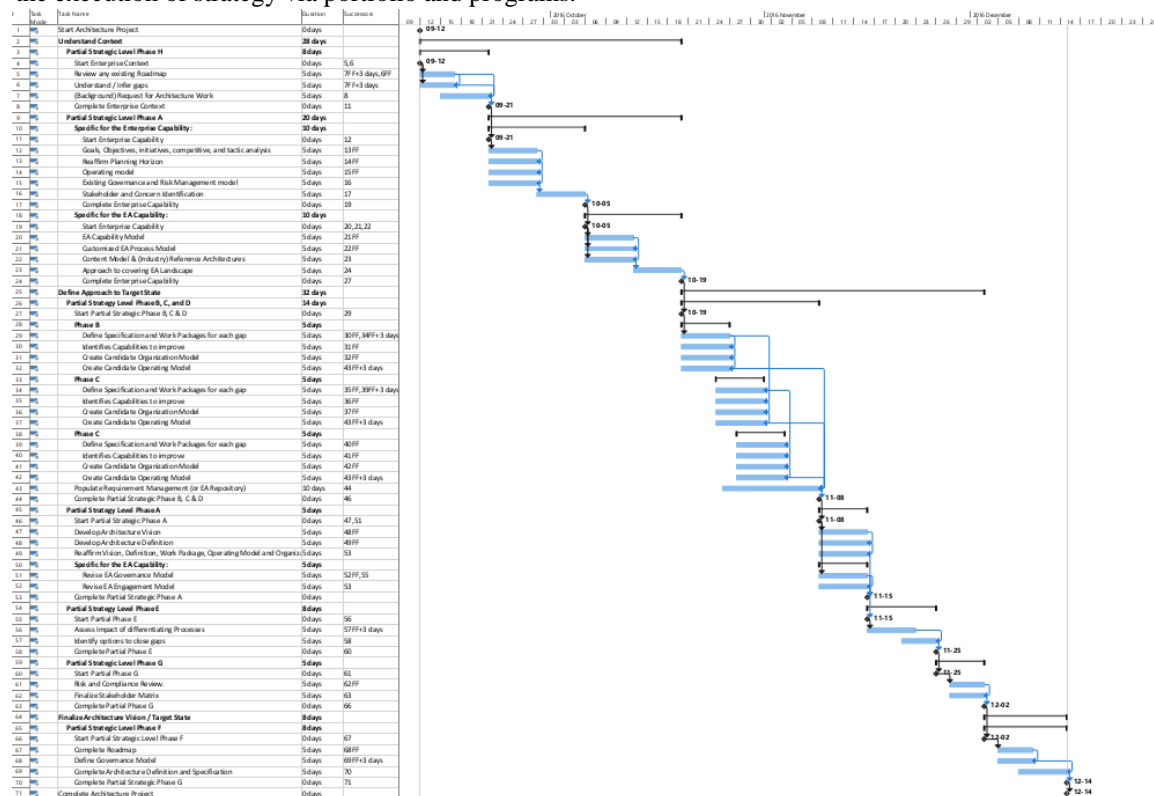
Figure 12: Sample Project Plan to Develop Architecture to Support Strategy

ADM Plan for Architecture to Support Portfolio

The path to developing an Architecture to Support Strategy is a configured journey through the ADM. This path follows this journey:

- Group work packages to themes

- Balance opportunity and viability

- Run up to budget

- Drive confidence of delivery

Figure 13 provides a sample project plan to provide Architecture to Support Portfolio. This project plan is explored in Walk Through Architecture to Support Portfolio (on page 78).

The processes iterate through the ADM to deliver an architecture that refers to a single portfolio.[25] The boundary and purpose of the portfolio are derived from the superior architecture. It will identify projects that comprise the portfolio. The project terms of reference and approach are identified. A key use is governing the execution of projects within the portfolio.

ADM Plan for Architecture to Support Project

The path to developing an Architecture to Support Strategy is a configured journey through the ADM. This path follows this journey:

- Ascertain dependencies

- Balance options and suppliers

- Finalize scope and budget

- Prepare for solution delivery governance

Figure 14 provides a sample project plan to provide Architecture to Support Strategy. This project plan is explored in Walk Through Architecture to Support Project (on page 87).

The processes iterate through the ADM to deliver an architecture that refers to a single project. The boundary and purpose of the project are derived from the superior architecture. The EA will identify discrete gaps and work packages that have been packaged into a project that delivers measurable value on the architecture roadmap. Further, the measures of compliance with the architecture are provided. Architecture for this purpose will create the Architecture Contract. A key use is ensuring value realization of the Implementation Project.

ADM Plan for Architecture to Support Solution Delivery

The path to developing an Architecture to Support Strategy is a configured journey through the ADM. This path follows this journey:

- Align implementers

- Guide delivery

---

[25] For the purpose of this discussion, this paper uses "portfolio" to refer a collection of projects that work to a common outcome. Whether a Practitioner's organization uses initiative, portfolio, program, or some combination will be determined by the organization's approach to change, how it has structured its PMO, and how the Enterprise strategy is structured. It is not in the scope of this paper to pursue the theoretical distinctions between appropriate use of these terms.

- Realizing the solution

Figure 15 provides a sample project plan to provide Architecture to Support Solution Delivery. This project plan is explored in Walk Through Architecture to Support Solution Delivery (on page 95).

The processes iterate through the ADM to deliver an architecture that facilitates solution delivery. (See Budget Control and Architecture to Support Solution Delivery for a discussion of the distinction between Enterprise and Solution Architecture.) This architecture is used to constrain how the change will be designed and delivered. It will clarify the purpose, gaps, and expected value that constrain all design and implementation. It will provide the controls and architecture requirements used to test conformance. It directly facilitates governance of implementation and operational change in the context of value realization.
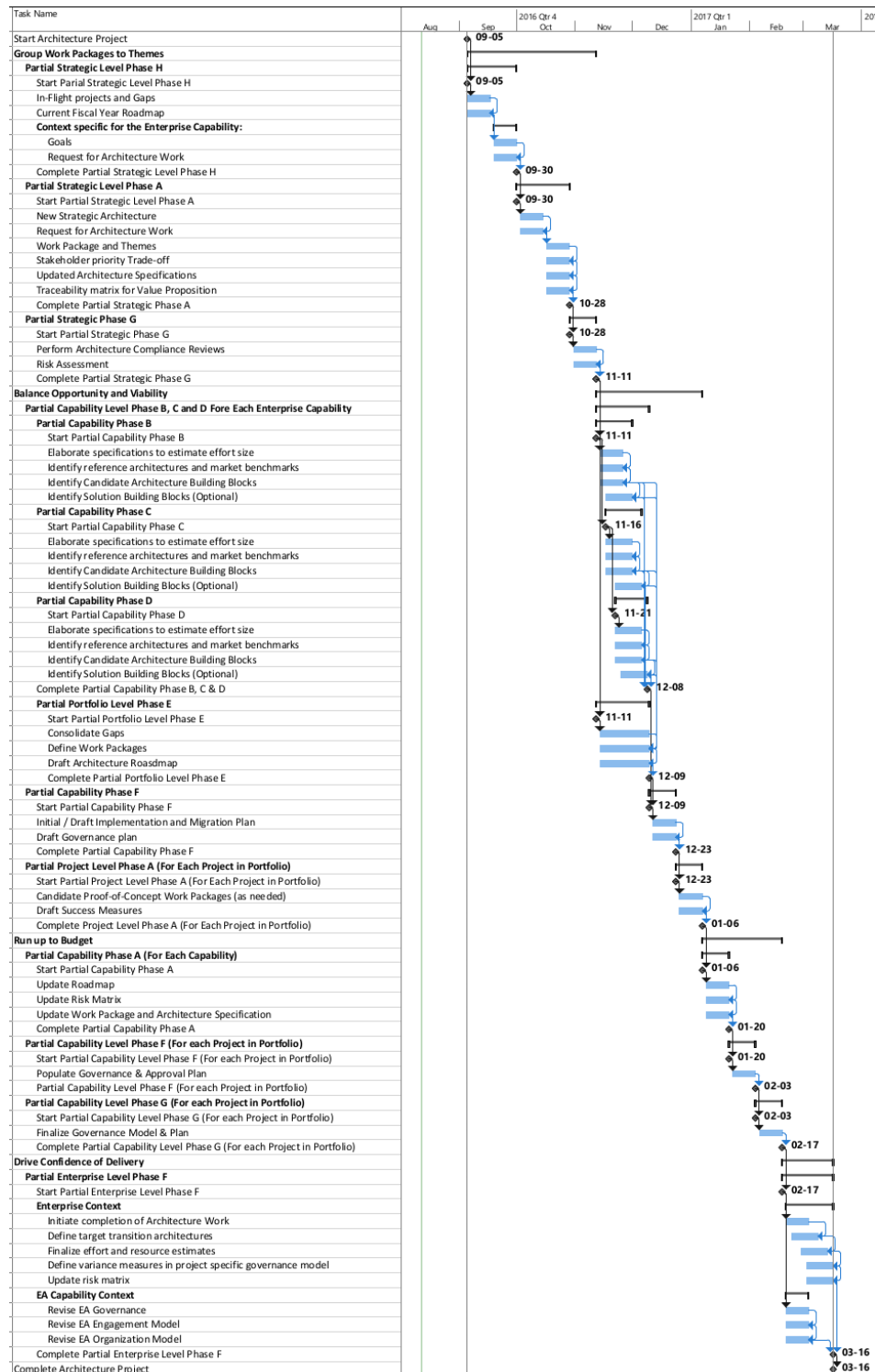
Figure 13: Sample Project Plan to Develop Architecture to Support Portfolio
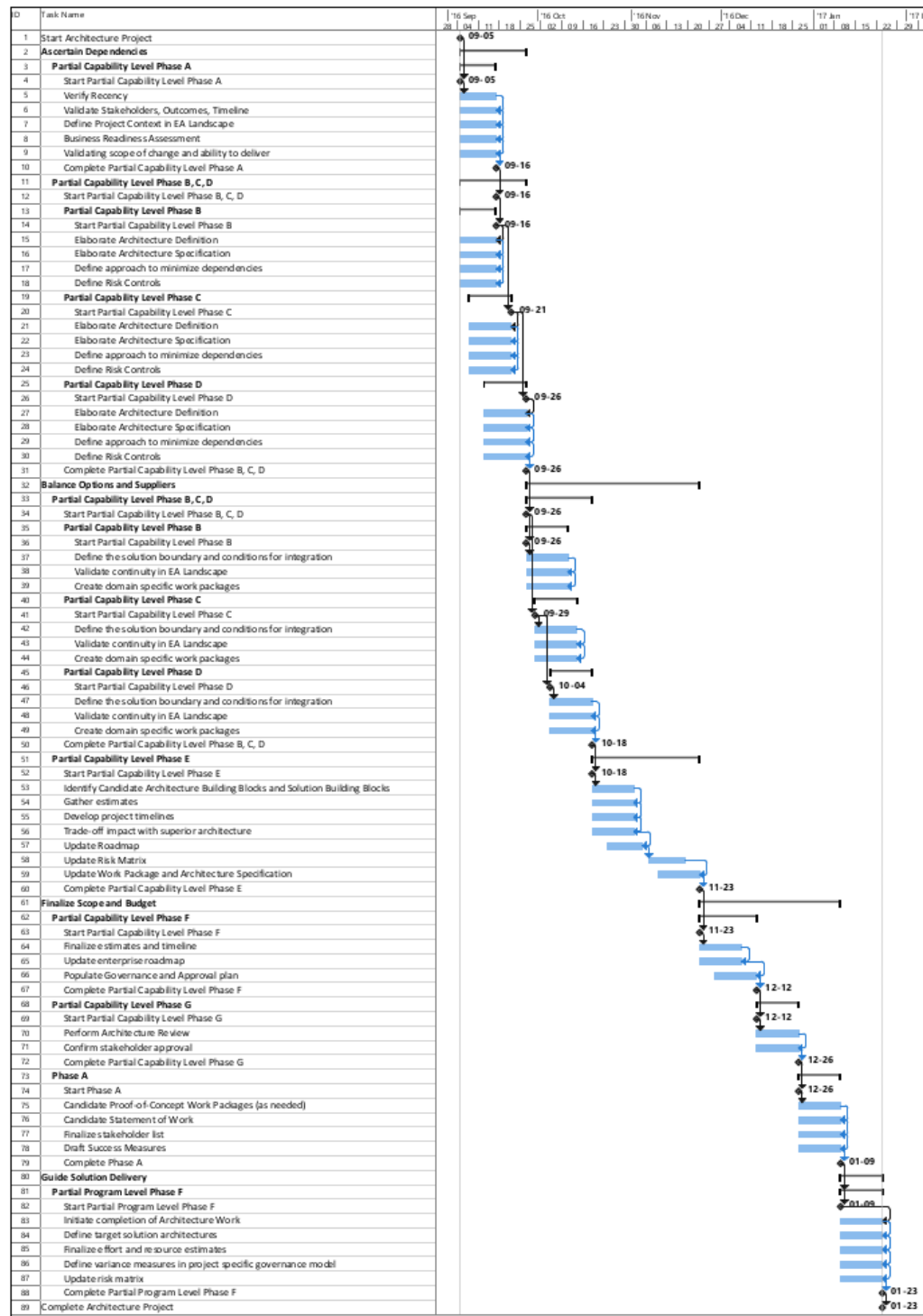
Figure 14: Sample Project Plan to Develop Architecture to Support Project
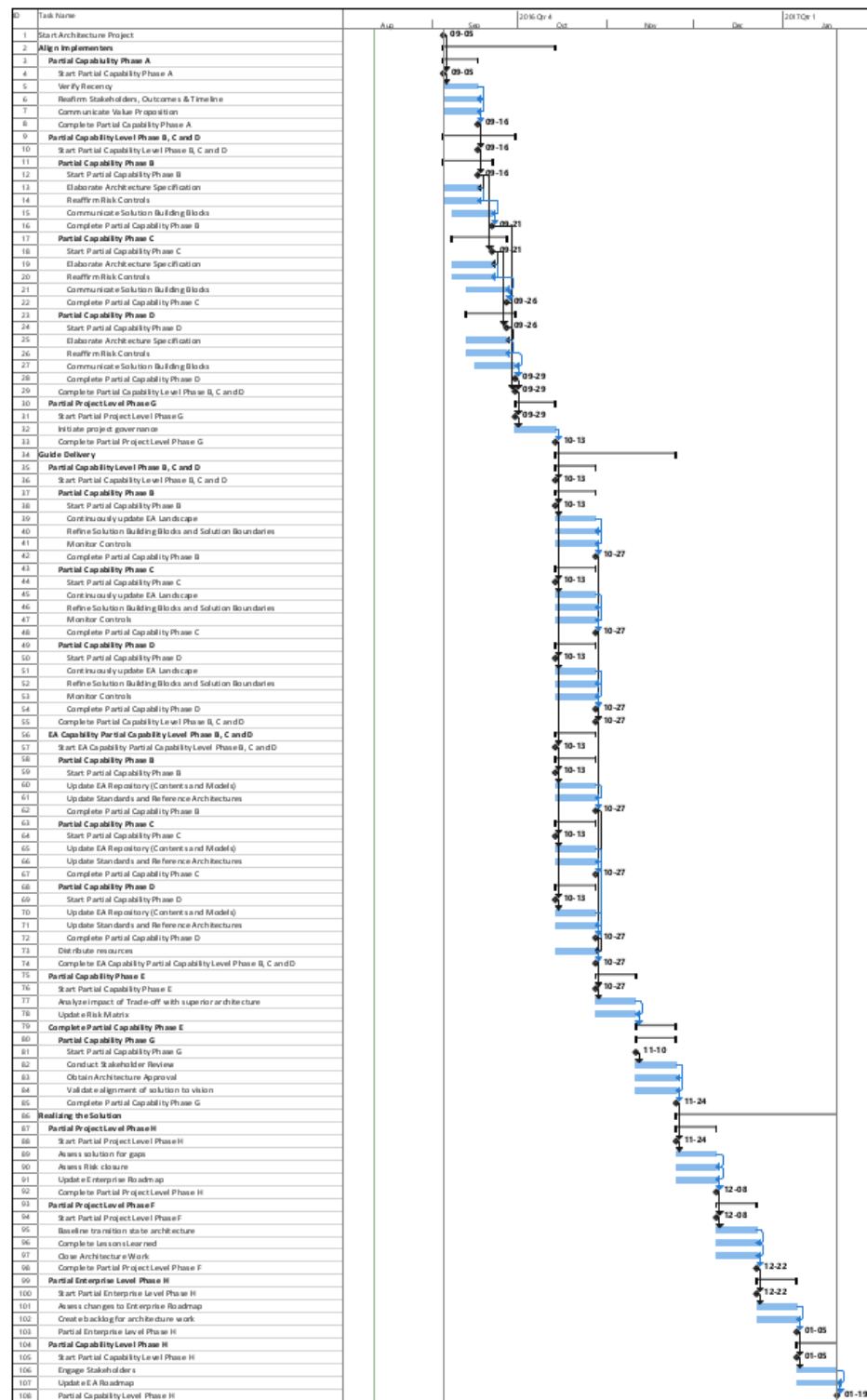
Figure 15: Sample Project Plan to Develop Architecture to Support Solution Delivery

Iteration Conclusion

At the start of this section, this paper suggested that you may understand the TOGAF ADM as a process model. If you did and continue to carry that notion, stop and think. The classic TOGAF diagram of the ADM is not an activity diagram. The TOGAF ADM is a logical method that places key activity steps together for the purpose of linking activity and information flow to produce specific outputs.

The important thing to realize is every time a Practitioner undertakes any activity within the scope of the ADM it is developing the contents of the EA Landscape. It is developing the EA Landscape through iteration. The phase being executed is the appropriate domain. If you remain stuck on trying to put the ADM in a one-pass linear order, you will draw bizarre looping phase diagrams. Think of the steps as a checklist.

Operating in the Context of Superior Architecture

The superior architecture always guides and constrains the development of more detailed architecture. As a quick summary, superior architecture is the less detailed approved target that overlaps in terms of breadth. This quick summary is complicated by the different states the superior architecture may actually exist in the EA Landscape.

The superior architecture may not perfectly align to detail, breadth, time-horizon, and recency. Further, the superior architecture may be in some mixture of current, transition, and target state.

Practitioners must treat the superior architecture as guides and constraints to current architecture development. Stakeholders have already approved the superior architecture in the EA Landscape; barring a material change, the Practitioner accepts prior work as cornerstones to build a current workaround.

Where there is a material change, both the current Architecture Project and the changes to the superior architecture must be properly approved and published through the governance process.

Managing Multiple States (Candidate, Current, Transition, and Target)

The Practitioner must track transition states across two characteristics: the first being time, and the second being a conformance test. Theoretically, it might be preferable to use transitions to track the value resting places and changes in conformance. Good practice is to architect to value resting states; a state where the Enterprise can receive value if all change activity is suspended. However, the pressure of the budget cycle forces us to use time is a pragmatic transition marker. Tracking to change in conformance facilitates the Implementation Project and operational change governance. To the extent possible, minimize transition states.

When considering transition states, the Practitioner needs to keep in mind the distinction between an Architecture Requirements Specification and an implemented system. Using the EA Repository as a CMDB confuses implementation record keeping and architecture. Practitioners have to keep in mind that many implementations or operational changes are not architecturally significant. See Architecture Governance (on page 117) for a discussion of the different roles involved in developing and using architecture.

Where are ABBs?

The TOGAF concept of the Architecture Building Block (ABB) is the effective Practitioner's friend. A good ABB facilitates time-to-market and completeness. As with most TOGAF definitions, knowing that an ABB is

"a constituent of the architecture model that describes a single aspect of the overall model" doesn't immediately tell us what they look like in an EA Repository.

An ABB will look like whatever it must be to describe part of the overall architecture – efforts to carefully define the contents and structure of this concept will founder on the variability and scope of what can be described within an EA Landscape. A building block is part of a greater whole that accelerates the effective description of the candidate architecture.

In some cases, it will be a re-usable description of part of the architecture; using it again enables the Practitioner to simply adopt a known successful way to address a problem. In this case, the ABB is complete in all regards, providing a complete description, and constraints that address repeated requirements. In other cases, it will not have the constraints and specifications predefined. In this latter case, the components of the description will be complete, but the detail will vary depending upon the requirements.

Part 3: Guidance on Developing the Enterprise Architecture

Approach to the ADM

The TOGAF Architecture Development Method (ADM) is the core of the TOGAF standard. This method sets the TOGAF framework apart from every other EA framework because it contains the "how".

The path through and around the ADM phases to develop architectures for different purposes is not simple nor linear. The level of detail and specificity of each architecture is different. For instance, to develop an Architecture to Support Strategy, all that is needed is to follow a path from Phase A through Phase D at the strategic level. Not all the steps are executed, but logical entities that drive Business, Applications, and Technology Architectures are captured and defined. Architecture to Support Strategy provides an end-to-end view of the Enterprise and a candidate roadmap to achieve target state. The governance model, as articulated in the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References), is leveraged to trace the rest of the architectures and their alignment to target state.

Key Activity

All architecture development has a set of consistent key activity that is essentially unchanged for different purposes.

Stakeholder Engagement and Requirements Management

The TOGAF framework places requirements management and stakeholder engagement at the center of architecture development. Practitioners develop EA in accordance with the preferences and priorities of their organization's stakeholders. Architecture is never sold to a stakeholder. stakeholder preferences are never manipulated.

Stakeholders own the architecture and the value preference and priority the architecture is expected to enable. Practitioners must completely submerge their preferences, biases, and priorities. Practitioners must act for their stakeholders.

This is one of the most difficult activities a Practitioner must perform. Good Practitioners are passionately engaged in the future of their organization, as well as participating in defining and realizing the target state. Practitioners typically perform several roles: they will act as Subject Matter Experts (SMEs) and agents for their stakeholders in addition to developing architecture – see Architecture Governance (on page 117) for a discussion of roles. As an SME, the Practitioner is a source of expert advice. As an agent, the Practitioner may speak on behalf of a stakeholder. In order to be successful when performing these roles, the good Practitioner must understand when they are acting in a different role and behave appropriately.

Effective requirements management is dependent upon clear traceability from the organization's vision, mission, business model, and strategies through the most detailed statement of requirement. In order to perform this, the Practitioner must carefully distinguish between direct effective support and loose association. Things that do not best enable the complete set of stakeholder preferences are distractions from the main chance.

When engaging with stakeholders, Practitioners must maintain the complete set of every stakeholder's preference, and the implications of those preferences. Success requires abandoning absolute and entering the realm of satisficing. Bluntly, if there is a single obvious best answer, the organization's stakeholders do not need an architecture.

Effective engagement is based upon effective communication. Effective communication is based on the concept of view and viewpoint. Different stakeholders have different concerns about the architecture. These concerns must be addressed and represented effectively to the stakeholder to enable the stakeholder to approve the Target Architecture (see Table 2, on page 29).

Trade-Off

One of the most valuable activities a Practitioner will perform during architecture development is facilitating the stakeholders' trade-off decision. Facilitating trade-off is often more valuable than finalizing an architecture description. Good architecture addresses complex problems. Complex problems[26] do not have clear, unambiguous best answers. Instead, they have reasonable compromises.

Trade-off requires a compromise between one stakeholder's preferences as well as between different stakeholders' preferences. Effective trade-off requires understanding value preference and priority as well as the scope of change necessary to realize the target.

As a rule, stakeholders underperform when that trade-off stands beyond their span of control or span of interest. In particular, stakeholders underperform when the trade-off involves the preferences of different stakeholders. Stakeholders typically overemphasize the institutional role and preferences of their portion of the organization.

Practitioners are most valuable facilitating trade-off between stakeholders and across organizational boundaries. This facilitation allows different stakeholders to effectively measure preferences, priorities, and costs that they do not intuitively understand. Best practice EA finds the best fit across competing preference, priority, and value. In facilitating the trade-off discussion, chase down all impacts and think through the end game needs. Work with the Enterprise risk management process to surface requisite dimensions. Think through all transition states. Leverage the architecture tool to handle the complexities of the EA Landscape and to accelerate the process.

Practitioners should not underestimate the value their organization receives from facilitation of trade-off across organizational boundaries.

Tourist Dashboard Decisions

This paper uses the term "Tourist Dashboard Decisions" as a shorthand for poorly analyzed choices. Controls and options in a car's dashboard can be difficult to learn. It delivers value (efficiency) over a period of time, and during critical scenarios. A kiosk empowers the visitor to navigate based on their interest and preferences; can present in multiple languages; enthrall the users with images of the attraction; and reduce the need for multiple backups as compared to humans serving as guides. Use of a kiosk presents its own challenges: need for constant power supply; ability to present a marked copy of a location map; and scale on-demand with traffic projections. However, consider that a tourist almost never visits the same attraction twice; the value of a kiosk lies in the learnability of the interface in seconds, as compared to the learnability

---

[26] Refer to Jeff Conklin's Wicked Problems & Social Complexity within Dialog Mapping (see References).

of a dashboard in a car or an aircraft. If we do not consider the value of rapidly clearing the queue of tourists, it would be easy to assume a design like a car's dashboard (assuming most people know how to drive a car – hence, operate a kiosk designed like a dashboard).

When a Practitioner is exploring the candidate Target Architecture and discovers what appears to be an obvious improvement without appropriate support or championship, they are making a "Tourist Dashboard Decision". When faced with such a circumstance, the Practitioner should look for the hidden value, as the learnability of the kiosk amplifies experience and potentially motivates the car's dashboard to change as well. Such values may never be described in normal terms of cost-benefit.

Phases B, C, and D – Developing the Architecture

Practitioners often find it surprising that the steps outlined in the TOGAF standard to develop architecture in Phases B, C, and D are identical. The steps are identical because the approach to developing an architecture, confirming the work product developed fits, and confirming approval are identical. These steps are also mandatory. Steps can be skipped, but the final outcome could be at risk.

What changes from purpose to purpose, domain to domain, project to project, and EA team to EA team is the level of detail, precision, and formality. All Practitioners should use the steps as a checklist.

Select Reference Models, Viewpoints, and Tools

Avoid rework. Practitioners test with the following questions:

- Given a set of stakeholders and concerns, what information do you need to know about the system being examined to address their concerns?

- Given a set of information, how will you model, represent, capture, and analyze it?

- Are there reference models that allow you to skip to gathering and analyzing rather than inventing?

- What information is missing from the EA Landscape right now?

Develop Target, Baseline, and Gap

Just enough for the purpose. If the current state is accepted, the only reason to describe the baseline is to develop gaps. If stakeholders, or SMEs, dispute the current state, especially its fitness to objective, then describing current state to get an alignment is useful. Otherwise, let us re-iterate: only to the extent necessary to determine gaps.

Consider the limitation of restricting description to where there is a gap. If part of the EA Landscape will have no change, and is not needed for traceability, what useful reason is there for a Practitioner to spend time describing it?

A recurrent question is how to describe the current state. Frankly, use the exact same techniques as the candidate. Description using the same technique at the same level of detail enables identification of gaps. A gap is simply everything that changes.

Identify the Work to Reach the Target

Without understanding the work required to reach the target, stakeholders will approve the impossible. Why wouldn't they want telepathy helmets and self-manufacturing products if they were free and easy?

The Practitioner is accountable for guarding value. A target provides an increase in value, at a cost of change. If you do not have an understanding of the work to reach the target, how can a Practitioner represent to a stakeholder that any target is a good idea and addresses the organization's preferences?

Resolving Impacts

Resolving impacts across the EA Landscape is one of the most important steps in managing the EA Landscape. The Practitioner explores the impact of their candidate architecture against other candidate architectures, transition states, the target state, and in-flight Implementation Projects. The Practitioner also works with the Enterprise risk management process to assess impact to the Enterprise's risk. Altogether, this is one of the most complex activities for an engaged high-functioning EA team. It requires a functioning EA Repository and solid analytic and reporting software. Every organization is a set of constantly changing interconnected parts. All architecture descriptions are approximations.

In practical terms, the more complex the EA Landscape is, the more difficult, and the more necessary, resolving impacts is. Practitioners attempting to manage an EA Landscape without an effective model and analytic tooling will struggle to resolve impacts. All impacts need to be resolved in terms of value expectation which is based upon clear traceability from the work required to realize the Target Architecture through the gap to the expected value.

Without care and attention to addressing the impacts across the architecture landscape in all of its states, the Practitioner cannot have confidence that their candidate architecture best serves the Enterprise.

Manage the information volume down to the minimum and constantly chase the minimum set of concerns that visibly support value in the eyes of key stakeholders.

Approval

Without approval by the stakeholders, no implementation governance is possible, and no governance of more detailed architecture is possible. Without approval, the Practitioner has a documented opinion. Stakeholders, SMEs, implementers, and decision-makers also have opinions.

Real approval is complex. Real approval should be complex. The Practitioner is assisting their organization select the best possible path against a set of competing preferences over time. They have taken the time to explore options and impacts.

With an approved Target Architecture, the future is defined, traceability to the objective is available, and trade-off has been performed. Good architecture trade-off explores options, cost, and benefits to reach the optimal answer for an organization. Often that answer is a compromise between competing interests.

Minimum Needed and Look in the EA Repository

Practitioners start and finish with the contents of the EA Repository.

Whenever analysis, or reporting, is needed, the first stop is the EA Repository. Practitioners should apply the following tests:

• Is the information that will address the question at hand already available?

• Is there a superior architecture that guides and constrains the task at hand?

• What is the minimum information needed to cover shortfalls in the EA Repository?

It does not matter whether the EA Repository is a well-structured modeling and analysis tool or a collection of presentations, start with the EA Repository. Gather and analyze the minimum to address the question at hand. Questions that do not have a clear line of site to understanding the system to address a stakeholder concern are beside the point. Good Practitioners are not paralyzed by the potential analysis that could be done; they perform the analysis that must be done.

ADM Conclusion

The TOGAF ADM sets the TOGAF framework apart from every other EA frameworks because it contains how to develop and use effective EA. It is not a simple nor a linear path around the ADM phases to develop the architectures for different purposes. It is, however, filled with tasks that are mandatory. Again, to skip tasks undertakes risks.

Walk Through Architecture to Support Strategy

Introduction

The objective of this architecture is to define an end-to-end Target Architecture and a roadmap to achieve it constrained by the planning horizon (normally three to ten years). This architecture will drive creation of several targeted change initiatives, define the boundary conditions for governance, and acceptance criteria for value reporting. Activities to develop this architecture will iterate the ADM at least once at the Enterprise level and once for the EA Capability.

On most occasions, EA initiatives are triggered in the middle of a business cycle. It is most likely performed by an Enterprise that has been operating for many years. A logical point to start the architecture work is to understand the rationale for EA work. Table 5 summarizes how the ADM phases are executed and to what outcome. The content of the table is discussed in detail in the rest of this section.

Table 5: Summary Table: ADM Phases and Architecture to Support Strategy

Topic	Mapping to TOGAF ADM Phase
Understand Context	Partial Strategic Level Phase H Enterprise context: • Review any existing roadmap • Understand/infer gaps • (Background) Request for Architecture Work Partial Strategic Level Phase A Enterprise context: • Goals, objectives, initiatives, competitive, and tactic analysis • Reaffirm planning horizon • Operating model • Existing governance and risk management model • Stakeholder and concern identification Context specific for the EA Capability: • EA Capability model • Customized EA process model • Content model & (industry) reference architectures • Approach to covering EA Landscape

Topic	Mapping to TOGAF ADM Phase
Perform Assessment and Analysis	Partial Strategic Level Phases B, C, and D Enterprise context: • Assess current and target operational levels for process, business terms, information systems (pplication, data, technology, etc.), and capabilities • Assess current and target levels for business and extended context, specific to the Enterprise • Identify candidate ABBs (optional) • Document and define the gulf between current and target Partial Capability Level Phases B, C, and D Enterprise context: • Assess current and target operational challenges, engagement with partners and suppliers • Organizational structure and stakeholder matrix • Reaffirm value proposition of the Enterprise Context specific for the EA Capability: • Revise EA content model • Revise EA Repository Partial Strategic Level Phase A Enterprise context: • Identify and analyze gaps • Identify viewpoints Partial Strategic Level Phase G Enterprise context: • Compliance review • Completeness and confidence assessment

Topic	Mapping to TOGAF ADM Phase
Define Approach to Target State	Partial Strategy Level Phases B, C, and D Enterprise context: • Define specification and work packages for each gap • Identify capabilities to improve • Create candidate organization model • Create candidate operating model • Populate requirements management (or EA Repository) Partial Strategy Level Phase A Enterprise context: • Develop Architecture Vision • Develop Architecture Definition • Reaffirm vision, definition, work package, operating model, and organization model for relevance Context specific for the EA Capability: • Revise EA governance model • Revise EA engagement Model Partial Strategy Level Phase E Enterprise context: • Assess impact of differentiating processes • Identify options to close gaps Partial Strategic Level Phase G Enterprise context: • Risk and compliance review • Finalize stakeholder matrix
Finalize Architecture Vision/Target State	Partial Strategic Level Phase F Enterprise context: • Complete roadmap • Define governance model • Complete architecture definition and specification

Understanding Context

Implicit roadmaps and direction have been used to execute the current year's initiatives. Most of them are meant to address a gap. Most likely the progress or the impact concerns triggered the need for architecture work. Document such concerns and initiatives as the draft Request for Architecture Work. Those concerns are probably valid even now.

When approaching Architecture for Strategy, achieving the goals of the Architecture Vision phase is arguably the most important step for achieving a proper rollout of the next phases of the ADM as well as setting the stage for success for subsequent architectures. An implicit constraint to developing the strategic architecture is the duration of planning horizon. The Target Architecture should be commensurate with the ability of the Enterprise to look into the future, competition, investment strengths, etc. Another aspect is the existing models for governance and risk management. It may not be defined or stated explicitly. It is the fastest path

to getting the efforts off the ground. If the EA Capability has not documented the model, spend the time to get it done.

The scope of a strategy architecture usually involves a wide breadth, a shallower depth, and a long timeframe. In order to define what is inside and outside the scope of the baseline and Target Architecture efforts, the following must be defined:

- The breadth, depth, and timeframe of the architecture landscape

- The level of detail to be covered in each of the architecture domains

- The partitioning characteristics of the architecture

- The known constraints

- The architectural assets to be leveraged, such as assets available elsewhere in the industry like frameworks, system models, etc.

As always, stay on top of what creates value for the Enterprise – meaning match the architecture to the problem at hand. The scope will limit the architecture to exactly what is needed to achieve the strategy goals and no more.

A key deliverable to this step is the creation of a Stakeholder Map which should clearly state the stakeholder concerns, requirements, and viewpoints as well as their classification and level of involvement. Other inputs from gaining an understanding of stakeholders are cultural factors, which can help the EA team understand how to present and communicate the proposed architecture.

This step is very important to strategy architecture since having a clear understanding of stakeholder needs, interests, visions, etc. will dictate how strategy architecture is understood by its sponsors and guide the EA team to act accordingly.

From a strategy perspective, it is important to ask whether the context of a business aligns with the mission. Do the capabilities match to the project scope? Are we carrying baggage from a previous project or from a different part of the company that is outside the confines of the architecture? Knowing the context of the work can help fine-tune the vision of the strategy architecture.

Finally, validate that the models specified by the EA Capability to analyze processes, engage with stakeholders, and deliver the architecture are relevant and current.

Assess the Enterprise

This is the core of the effort required to deliver Architecture to Support Strategy. Working across the breadth of the Enterprise, identify, define, and articulate as clearly as possible the operational state. This analysis covers business processes, information systems, technology, business terms, security, service providers, customer satisfaction, etc. For each of these, gather the desired operational state that would enable the Enterprise to achieve most or all of its objectives.

Completing the assessment may require use of techniques like Strategy Map or Five Forces. The outcome from such exercise will change the strategy statements and objectives. When the initial analysis does not provide the growth amplification expectations of the stakeholder, employ these techniques to guide the stakeholder to explore new ways to play in the market. The architecture being delivered is driving a change,

but the analysis is just a path to identify a right change to introduce. Some or all work products created while developing the architecture may not go into the Architecture Repository or become a deliverable.

The assessment should be performed to address key concerns of the stakeholders. If the Enterprise is chasing agility, assess for current and desired[27] agility levels. If it is after operational stability, assess current and desired. If the need is the ability to replace suppliers with ease, assess it. It is perfectly acceptable to state that one or more capabilities or information systems or processes are not needed in the desired state. Likewise, it is acceptable to move a capability or service from being a differentiator from competition to "on par" with competition. These are indirect statements of direction the Enterprise is planning to take. Validate that the value proposition, objectives, and the assessment values for the desired state are consistent.

What the Enterprise is after is defined in the context and Request for Architecture Work. It is likely that stakeholders may state new concerns to be assessed. Refine and finalize the Request for Architecture Work after assessments. Remember that the goal is to capture just enough data to identify the gaps. How the outcome of each process, application, service, or capability measures against the concern is sufficient to complete the assessment. Going after who made the application or what version is deployed in the data center are noise and should be avoided.

The chasm between current state and desired state is the chasm the Enterprise has to cross to achieve its objectives. The chasm has to be acknowledged and agreed upon by all stakeholders.

In order to communicate what concerns were assessed across what capabilities, processes, information systems, etc., identify appropriate viewpoints. Validate that the team performing the assessment followed the documented EA processes and consulted requisite and relevant SMEs and stakeholders.

In order to provide confidence to the stakeholders of the completeness of analysis and resultant development of the target state and roadmap, have a detailed trail of the personnel consulted. Employ any of the standard techniques like interviews, surveys, inspections to gather the current and target state information. For each of these techniques, there are well researched metrics for the number of stakeholders and SMEs to be consulted. Completeness and confidence in the assessment is the Achilles heel of this architecture.

Define an Approach to Target State

With all the data gathered, look at the whole picture: where the Enterprise wants to go, the forces acting on the Enterprise from outside and within, resources it possesses, and finally the structural and behavioral changes needed. Each providing new specification. Each refining the view of the gaps. Some of the requirements may be not vetted against the desired state. As long as it is not in violation of the desired state and the objectives, it is a candidate that needs to be recorded.

---

[27] Use of the term "desired" is intentional to communicate the fact that it is difficult for a human to foresee and consider change parameters in the future. Until a consensus is reached across key decision-makers, data gathered during assessment is an opinion or a wish. Once confirmed, it becomes a candidate target state. Once funded or signed off, it becomes the target state.

An architect adds most value in correlating the facts, and identifying a potentially new operating model, organization model, and capabilities the Enterprise should invest and improve upon.

Confirm Enterprise Change Attributes

This step looks at how to implement an architecture taking the organization culture into consideration when assessing the business units and overall Enterprise in terms of their transition capabilities and skill sets. These assessments should be documented in an Implementation Factor Assessment and Deduction Matrix so that it can be used as an archive and record of decisions taken. Culture is very important to strategy architecture since strategies are long term, and often culture is set for the long term. Getting these two in sync is paramount to building a successful architecture. Other components of this step that are relevant to the strategy architecture include assessing the context that shaped the need for the strategy and performing a gap analysis of the Architecture Vision to the candidate architecture.

Develop Value Proposition

It is important that not only the value proposition for strategy architecture be understood by stakeholders but also the effort needed is accepted in its entirety. Consent and understanding should be manifested in a simple solution concept diagram that illustrates the major components of the solution and how the solution will positively impact the business. Since the value proposition is specific to stakeholder interests and concerns, it is important to pay close attention in this step as well-defined value propositions are key to strategy architecture success. For any architecture, sub-steps involve:

- Risk Assessment – leverage risk management processes to determine the level of risk appropriate to the vision.

- Determine Value – link value to work packages as they pertain to stakeholders or stakeholder groupings.

- Determine Key Performance Indicators (KPIs) – can be associated with concerns, risk assessment, and value. Determining the KPIs is necessary in the strategy architecture in connection to governance.

Determining the value proposition and how it is linked to various stakeholders and deliverables will help formulate very high-level definitions of the baseline and target environments from multiple points of view. Strategy is all about high-level concepts, but agreement on these concepts is key for a successful vision to be formulated and adhered to.

Identify and Sequence Work Packages

Logically group the various activities into work packages. This way the missing business capabilities can be assessed and, in the solutions column, proposed solutions for the gaps and activities that might orient towards a new development can be recommended. This step allows us to prepare for solution delivery, as the new developments might already hint at using external service providers.

Having done the sequencing and sifting down to relevant architecture requirements, the candidate roadmap and candidate Target Architecture are ready to construct the Architecture Vision. Create the initial version of the roadmap by consolidating the work packages from the previous steps while keeping in mind that this roadmap will link to subsequent phases. At the broadest level, the roadmap should define where the business wants to go, how it will get there, and by which means. Keeping an eye on the sufficient level of detail needed for this roadmap to be implemented should forbid the architecture to transition to different results.

Finalize Architecture Vision and Target Architecture

Tie-up any loose ends or mismatch in work packages and capabilities; resolve the impacts to the candidate architecture, and resolve impacts across the Target Architecture by performing stakeholder concern trade-off analysis. The roadmap should be significant in breadth for clear outcomes but shallow enough in depth to outline work packages without going into too much detail. The transition and migration plan must likewise demonstrate a minimum activity necessary to realize the roadmap. It is key to take the context of the Enterprise into account when formulating the implementation plan since there will be different approaches to consider depending on the business.

Sub-steps to follow for both of these points include:

- Context Assessment – assess the roadmap components and work packages in the context of the capability, value, and risk assessment.

- Describe Candidate Transition Architecture – where there are significant points being changed in the Target Architecture along the roadmap, create a transition architecture that supports new models, identify building blocks to be used in the transition, identify views that address stakeholder concerns, and identify specifications.

- Resolve Impacts Across the Architecture – determine the impact and interact with risk management to create a plan for the transition.

- Perform Trade-off Analysis – interact with the requirements management process to update requirements and with risk management to update risk based on these trade-offs.

- Have the Target State Approved by the Appropriate Stakeholder(s) – you do not have a roadmap until the organization has signed up to do the work. Without an agreement to do the required work you only have an intention to change.

Conclusion

Communicate the Architecture Vision and populate the governance model and process with stakeholders, review cycle, and objectives. Ensure that stakeholders and decisions-makers understand, agree with, and provide the license to proceed with populating the EA Landscape. This license to proceed with the stated vision, Target Architecture, and the roadmap constrains and guides all future architecture work. Creation of a value chain, strategy map, or balanced scorecard can be completed meaningfully when the Architecture to Support Strategy is ready.

A list of duplicative efforts that require rationalization and a graph of sustain and improvement capabilities are populated into the roadmap. The stakeholders have successfully directed the creation of the architecture

and have populated the governance details for further detailing and implementation of the architecture. This is the superior architecture[28] that will guide and direct the Architecture to Support Portfolio.

Success is measured by alignment on the target state and clear understanding by the decision-makers and stakeholders of the effort required to achieve the target state.

---

[28] Superior architecture is an architecture that constrains, guides, and directs population of the EA Landscape within the scope of the Request for Architecture Work. Architecture to Support Strategy is the superior architecture for Architecture to Support Portfolio. Architecture to Support Portfolio is the superior architecture for the Architecture Project. The Architecture Project is the superior architecture for Architecture to Support Solution Delivery.

Walk Through Architecture to Support Portfolio

Introduction

Almost all EA engagements, external or with an in-house EA team, are initiated for an Enterprise that has been in existence for a while. Whether explicitly initiated or acknowledged, an architecture is in place and solutions are being delivered against that architecture. Even when the Architecture to Support Strategy has been created for the first time, there are ongoing efforts and their impact that will have to be accounted for.

The primary objective of Architecture to Support Portfolio is to identify projects, identify dependencies and synergies, and prioritize and initiate the projects. From that perspective, it would appear that all of the work is confined to Phase F to complete the architecture work and transition to solution delivery work.

The Enterprise's solutions are delivered on a continuum. This continuum is split into four phases, all focused on achieving the objective to meet stated goals. These phases are:

- Stay on par with other players in the market for a given capability

- Maintain the edge a capability has over other players

- Create new differentiations in capabilities

- Create new markets and revenue streams

Once a new capability or a differentiation in a capability is achieved, the incremental advantage will have to be maintained.
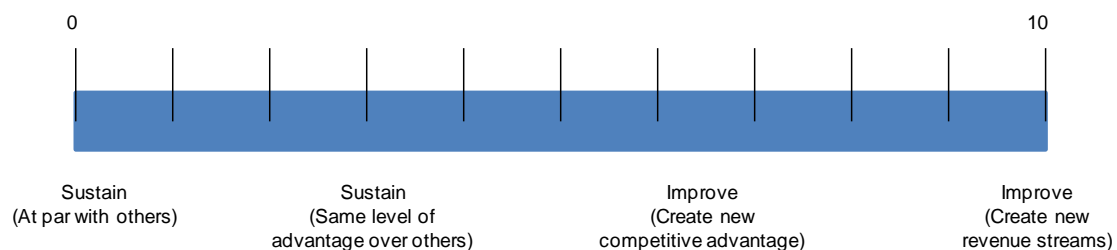
Figure 16: Capability and Project Continuum

It is imperative that Architecture to Support Portfolio takes into account an existing implicit or explicit Target Architecture and the impact driven by in-flight projects. Hence, in true sense, this work starts in Phase H of the ADM. The work is considered complete when all the specifications that constrain the Architecture to Support Project are defined, understood, and signed off. In other words, the need to perform Phase A for the solution delivery projects that are triggered by the portfolio is complete.

In doing to so, the architecture provides a data-driven approach to reduce the possibility of one set of decision-makers netting the majority of the available budget because of the way it has been in the past. This is achieved by developing appropriate models, like-to-like comparison, and incremental exploration of the EA Landscape to assess impacts and dependencies.

It is imperative that the Architecture to Support Portfolio concludes at least 30 days before the budget preparation. A best practice is to is offset this work by at least a quarter (three months) from the business cycle of the Enterprise.

Questions answered by this effort are:

- Is the architecture recent and current enough to guide decisions?

- What is the confidence that the allocated budget drives the Enterprise closer to target state?

- Are the controls on risks sufficient enough to trigger and guide viable alternate actions?

- How often is the solution delivered to be inspected to assure general correctness of direction?

- How to identify and initiate changes when any of the trade-off criteria are impacted?

When pivoting on program and project management concepts, a portfolio can include operational improvement efforts; not a clearly defined end-date for closure. The intrinsic value of the Enterprise is elevated when related and cohesive parts of the EA Landscape are improved. From an EA point of view, a portfolio addresses improvement of the intrinsic value and reduction of risk factors.

Table 6 summarizes the activities and use of appropriate steps from the ADM phases. The content of the table is discussed in detail in the rest of this section.

Table 6: Summary Table: ADM Phases and Architecture to Support Portfolio

Topic	Mapping to TOGAF ADM Phase
Group Work Packages to Themes (on page 81)	Partial Strategic Level Phase H Enterprise context: • In-glight projects and gaps • Current fiscal year roadmap Context specific for the EA Capability: • Goals • Request for Architecture Work Partial Strategic Level Phase A Enterprise context: • Updated strategic architecture • Updated roadmap Context specific for the EA Capability: • Work package and themes • Stakeholder priority trade-off • Updated architecture specifications • Traceability matrix for value proposition Partial Strategic Level Phase G Enterprise context: Perform architecture compliance reviews Risk assessment

Topic	Mapping to TOGAF ADM Phase
Balance Opportunity and Viability (on page 83)	Partial Capability Level Phases B, C, and D For each capability or project in the portfolio: • Elaborate specifications to estimate effort size • Identify reference architectures and market benchmarks • Identify candidate ABBs • Identify Solution Building Blocks (SBBs) (optional) Partial Capability Level Phase E For each project in the portfolio: • Identify solution providers • Readiness assessment • Gather estimates • Assess viability and fitness of solution options Partial Capability Level Phase F For each capability in the portfolio: • Initial/draft Implementation and Migration Plan • Draft governance plan Partial Project Level Phase A For each project in the portfolio: • Candidate proof-of-concept work packages (as needed) • Draft success measures
Run Up to Budget (on page 84)	Partial Capability Level Phase A For each capability or project in the portfolio: • Update roadmap • Update risk matrix • Update work package and architecture specification Partial Capability Level Phase F For each project in the portfolio: • Populate governance and approval plan Partial Capability Level Phase G For each project in the portfolio: • Finalize governance model and plan
Drive Confidence of Delivery (on page 85)	Partial Enterprise Level Phase F Enterprise context: • Initiate completion of architecture work • Define target transition architectures • Finalize effort and resource estimates • Define variance measures in project-specific governance model • Update risk matrix Context specific for the EA Capability: • Revise EA governance • Revise EA engagement model • Revise EA organization model

Group Work Packages to Themes[29]

The minimum dataset required to initiate this effort is:

- Current fiscal year's roadmap (to the extent available)

- List of in-flight projects and relationship to objectives

- Strategic architecture (gaps, work package, and candidate roadmap) for the next fiscal year, from Architecture to Support Strategy

- Catalog of stakeholders, decision-makers, and implementers

- Risk catalog

(Note: The backlog from the current fiscal year is not of concern, as the Architecture to Support Strategy has accounted for them.)

Given the context surrounding the Enterprise and the EA project, develop a Baseline Architecture from the current state architecture created by the superior architecture (Architecture to Support Strategy). The Baseline Architecture is not a physical thing. It is a point of reference in time, defining a metric and a measure to enable value reporting. The baseline is a collective view that provides credit for value added by in-flight projects. All value assessment and trade-off shall be performed against the baseline.

The Baseline Architecture groups the in-flight projects against the new objectives defined in the Target Architecture. It is possible that in-flight projects may not align cleanly with the Target Architecture. When a project aligns to more than one objective, assigning credit from such an effort to all objectives to create the baseline will not impact the value reporting. The impact of gaps between current state and Target Architecture will invariably outweigh.

Using the Architecture Vision as reference and the list of work packages, develop a set of themes, if not previously defined by Architecture to Support Strategy (prior architecture work). It may be necessary to create multiple baselines, one for each theme. Themes are defined by factoring the current and target organizational structure, productivity, differentiation, and scaling objectives. The organization structure

---

[29] Terms like "initiative", "portfolio", and "program" carry organizational connotations and often derail us from communicating the message. Most of the definitions derive from investment management concepts, which essentially states portfolio as a mix of assets that matches the objectives balancing risks against performance.

As defined by the Project Management Institute: "A *portfolio* is a collection of programs, projects, and/or operations managed as a group. The components of a portfolio may not necessarily be interdependent or even related, but they are managed together as a group to achieve strategic objectives." And: "A *program* is a group of related projects managed in a coordinated manner to obtain benefits not available from managing them individually."

According to Robert G. Cooper: "Portfolio is a dynamic collection of new and existing product or service development efforts, to allocate, de-prioritize, or regroup resources in response to dynamic opportunities, multiple goals, and strategic considerations, interdependence among projects, and multiple decision-makers and locations."

All of these definitions do not explicitly address the continuity and connectedness of the efforts in the context of an Enterprise. In order to stay away from such limitations, this paper resorted to using "theme" to indicate that work packages should be grouped in such a way as to enable populating neighbors in the EA Landscape. One theme may populate the Operational Excellence capability landscape while another may populate the Financial Controls capability.

articulates stakeholders, decision-makers, and implementers, their interests and concerns. As the work packages are moved across themes, perform an assessment of impact to stakeholders, decision-makers, and implementers. The resulting grouping of the work packages can be suboptimal due to dependency on pending organizational change.

When performing EA activity for the first time in the Enterprise it is safe to assume that there were no target transition architectures that were used to create projects in the current year. The Target Architecture and gaps were inferred by whoever drove the budget preparation and budget allocation. Many of the in-flight projects could have a target completion date that extends beyond the next couple of business cycles. Altering the course of these initiatives takes time and, hence, suboptimal architectures in the first go around of the architecture effort. Revisit the gaps list created by the Architecture Project and work packages, and make appropriate adjustment due to in-flight projects and any inferred roadmap for the current fiscal year. Prioritizing, estimating, and sequencing of this list is the scope of work for the Architecture to Support Portfolio.

To identify the prioritization of the effort, build Table 7. Populating the table forms the basis for performing further elaboration of the EA Landscape. Any cell in this table without data conveys that the architecture is not complete.

Table 7: Work Package Grouping

Portfolio Theme	Work Package Name	Work Package Required	Importance	Impact Realization Timeline	Effort Required	Magnitude of Investment

The importance of a work package is carried over from the strategic architecture. The last three columns will be populated as the architecture is developed further. As noted earlier, the work package to "theme" association is made using the lens of improving intrinsic value of the Enterprise, populating cohesive parts of the EA Landscape.

Analysis of the mapping between portfolio themes and stakeholder concerns identifies the subset of stakeholders to engage for each portfolio. For each portfolio, reaffirm that there are no changes in the internal and external forces that created the work package. Identify resources required and track the resources that cross organizational boundaries. It is typical for most organizations to require an elaborate process to move resources. Identification of such a need changes the dependencies and priorities of the work packages.

Using the stakeholder concerns from prior architecture work and the new grouping of the work packages, perform a trade-off analysis to quantify the changes to gaps and cascading impact on time to achieve the target state. Identify any new risks and develop appropriate controls. Using Table 7 and the mapping of work package to objectives via gaps, reaffirm that the value proposition delivered by the portfolio is aligned to the objectives.

The work packages carry an attribute to identify whether they are new or a carry-over from the current fiscal year's effort. From now on, the merits of the work package in shortening the path to target state drives decisions to invest. Continuation of the current efforts may be factored in, but they are not a determining factor. Now, a reasonable candidate Architecture Vision for each theme, and hence, a portfolio is created.

Balance Opportunity and Viability

The analysis and architecture development so far has been heavily focused on an inside-out approach. It is time to seek help outside the Enterprise. For the kind of changes being driven, potentially accelerating solutions might be available in the market – within the same industry vertical or otherwise. Technological developments and environmental changes might present new options to meet the needs of the work packages. Considering business cycles of suppliers, partners, and the Enterprise, it may be prudent to initiate identification of implementers now. These implementers are not decisions-makers or stakeholders. It is not good practice to include them in the stakeholder matrix.

Develop the Business, Information Systems, and Technology Architecture specifications to the extent needed to scout the market for options. The focus is more on identifying the motivations behind the solutions than identifying a solution. If the purpose is to transmit information digitally, identify whether imaging is not an acceptable option. This still leaves the option to innovate, if needed, the right fit at the solution delivery stage. A related question would be: is the transmission of data for record-keeping purposes or transaction management purposes? Such a motivation identifies attributes of the building blocks and potential reuse of solutions already employed in the Enterprise. Assess the solution options more from an exclusion point of view, rather than narrowing down to "the solution".

In elaborating the architecture, new risks and dependencies will arise, and so should appropriate controls. Develop a matrix of options, risks, and controls to enable viability analysis and trade-off with stakeholders. Keep populating the requirements management function with data from such elaborations. Identify the list of standards and reference architecture that can be leveraged or imposed as limiting conditions on the solution. Identification of such standards and architectures amplifies and drives specificity of the (constraints) architecture specification from the superior architecture. It may also provide an accelerated path to solution. Capture all possible attributes to inform trade-off analysis.

It is time to reach into the EA Repository for viewpoints, views, appropriate building blocks, and reference architectures to develop an approach to address the gaps. The viewpoints should provide a point of reference to the EA Landscape that is relevant for the stakeholder and decision-maker. Continuously validate that specifications for all work packages in the theme are elaborated equally, to the extent possible and necessary to decide the priority and resource needs.

Identify pockets where a solution may have to be invented. In such a case, create new work packages to perform proof-of-concept validations before scaling out. Understand that proof-of-concept work is actually implementation, not architecture. Architecture work is identifying the placeholder required to allocate appropriate funds and mitigate unknowns. The main focus of the Architecture to Support Portfolio is to maximize the mileage gained with available resources. The second objective is to identify conditions under with projected mileage gain is achievable. The third is to identify barriers to achieve the goal and build efforts to diminish the impact of such barriers. The final objective is to provide assurance of investment to reward ratio being unaltered. Populate the list of projects required to meet these four objectives.

Gather effort and resource estimates for all work packages. Revisit the dependencies across work packages. Identify the importance and impact of the work packages. The ability to authenticate the identity of the person carrying a ticket will vary with context. An Enterprise may have the same need for more than one scenario or portfolio. Or, in the case of boarding an aircraft, multiple agencies may have to be involved. Such work packages have high importance and impact, requiring early investments in the overall improvement cycle.

Perform an opportunity analysis factoring viable options to approach the solution. Remember, the focus is driving a baseline estimate and assurance of achievability of the target. The validation of the portfolio and the trade-off is focused on grouping by theme, related impact, and importance assessment. The decisions driven here impact the distribution of limited resources across the investment continuum.

Run Up to Budget

Internal Engagement

Other than line of business leaders, personnel from the office of the financial controller and Project Management Office (PMO) are key to driving the budget. The objectives of these two teams are fundamentally different, but converge once a year – the time of budget preparation. The convergence is around the trend on variance to budget. Enterprises develop guidance on year-on-year funding and budget trend based on statistical data, without any qualification for the value delivered. It is normal for the delivery or execution teams to ask for more than is needed or to keep the same level of ask, without sufficient demand, for fear of losing funding.

Another factor that could arise is the conflict due to gaps in the agility expectation of the service consumer (say sales team) and that of the service provider (say licensing and pricing team). Such a conflict creates duplication of capabilities and service in the guise of a different objective or effort name. Preparing for the budget, the EA team works to eliminate variations from such "opinions" or "duplications" of the past using gaps and work packages.

It is highly likely for the superior architecture to recommend organizational changes as well. In this case, the Human Resource (HR) team is going to play a more critical role in budget preparation than ever before. It is not the responsibility or the function of the EA team to drive decisions. EA has to frame the conversation and the directions to identify the right resources to lead and drive change. It is imperative that the engagement of all concerned internal teams – mainly HR, PMO, and finance – is key to the success of delivering the Architecture to Support Portfolio.

Has the Target been Reached?

Having driven confidence in reducing sources of artificial variance to budget, next to tackle is accuracy of the estimates. When the changes require a reasonable number of proof-of-concept efforts to be done or require employment of specialized services, veracity of the estimates would be questioned. In order to drive the level of confidence, it would appear that more time, more analysis, or more iterations are needed. Other than time, here is a short checklist that will indicate that it is time to stop iterating:

1. For each "theme", have the work packages been classified into a capability continuum (a work package cannot address both Sustain and Improve)?

2. Are the dependencies and cascading impact of work packages acknowledged by decision-makers and implementers?

3. Is there a contiguous elaboration and exploration of EA Landscape?

4. Have the mitigations and controls for risks (unknown events) been added to the portfolio?

5. Is there a blend of operational excellence and fitness for purpose within each theme?

6. Are there any recency concerns?

7. Is a raw estimate and contingency factor available (% buffer to account for market and external trends)?

8. Is the ratio of growth in breadth of coverage architecture specification to depth of coverage diminishing between iterations?

9. Is the variation in estimates between current and previous iteration less than the contingency factor?

10. How many of the efforts are one-time executions to support transformations?

The point of diminishing returns is met when positive responses are given to either (8) or (9) above. Mostly during the first two to three years after initiating an architecture-driven planning cycle, the EA team will run out of time before (8) or (9) could be met. Plan for recommending a discretionary spending bucket.

To complete the architecture work, update the architecture roadmap, risk matrix, architecture definitions, and specifications to the extent needed and necessary. As needed, consult and conduct reviews with SMEs and stakeholders to validate the direction. For each theme, define the governance plan and model that is acceptable to stakeholders and decision-makers.

Drive Confidence of Delivery

Useful architecture drives change and simplifies decision-making. The objective of budget preparation is to drive confidence of estimates, confidence of delivery against the roadmap, and garner the resources required to drive change. The set of prioritized work packages grouped by themes that traces to objectives drives confidence in responses to the "why" and "what" questions. The set of estimates that is backed by variance control drives confidence to the "how" and "how much" questions. Creating a set of project governance that reduces the chances of execution decisions delaying the time to target state serves the final objective of this architecture – balancing innovation and considered controls.

Develop just enough views, models, and specifications to support the budget request. These documents are supported by a matrix of accountable parties for delivery and accountable parties for acceptance, usage, and sign-off. Success measures are articulated in value terms – controls in cost measures, and risks and outcome in value measures.

Initiate activities to complete the architecture work. This involves populating the appropriate project vision documents, project architecture definitions, project stakeholder list, communication plan, and conditions that govern trade-off. Populate the data required by monitoring the system for each project, should the project be approved for execution. Populate the dependency matrix in accordance with the boundaries set for each project and the "theme". The Architecture Project cannot be completed until the Architecture to Support Project is delivered. Initiating the effort at this stage communicates the decisions at the strategy level that can be revisited in the future. The last validation is to define that the operating model (recovery-driven or engagement and continuity-driven) is aligned to the business model.

Request for Architecture Work Originating from a Random Idea from the Wild

In a well-run, creative organization many good ideas are not derived from gaps identified in the architecture. In these organizations, a Request for Architecture Work comes from someone with a good idea for improving the organization.

With a request from the wild, the Practitioner will typically engage with a strong champion and identify holes in the EA Landscape. There is little need to worry about bumping shoulders with other identified gaps and work packages. However, the champion often will have a limited, or myopic, view of the stakeholder's preferences and concerns.

The Practitioner must take care to stay within the context of the wild architecture development relying on the mission, vision, and strategy of the Enterprise. Requests from the wild should be expected to challenge the *status quo*. The inherent creativity is welcomed by good Practitioners. Without much guidance from the strategy or portfolio to constrain the architecture development, Practitioners must ensure that identification of the correct stakeholders is completed and that the concerns reflect the stakeholder's preferences and priorities – see Phase A: The Starting Point. Not all champions are stakeholders, and all Architecture Projects are subject to superior architecture.

There is a need for critical thinking around the preparation required to insert the architecture developed in response to a receipt of a Request for Architecture Work from the wild at the optimal point in the sequence of work within the Enterprise's roadmap, or implementation plan. Well executed, the organization is able to balance creativity and innovation with the benefit derived from clear understanding of dependency to value realization.

While most Requests for Architecture work from the wild are for Architecture to Support Project and Architecture to Support Solution Delivery, strong champions will drive a portfolio initiative.

Conclusion

Conduct periodic value assessment and reporting to communicate lessons learned and whether the portfolio created is delivering organic change, radical innovation, or maintains the *status quo*. Implementation Projects deliver value a few quarters after the project is closed. It is the responsibility of those managing the portfolio to track and report value. Add to the portfolio an explicit backlog item to monitor and report value realized.

In the event this architecture is supporting a merger, acquisition, or divestiture activity, include explicit recommendations to tackle the impact of technology in easing the business operations, asset, and risk accounting.

Success is measured by alignment by the decision-makers on a number of concurrent streams, total resources required over the planning horizon period, and trade-off criteria.

Walk Through Architecture to Support Project

In this context, the architecture is used to clarify the purpose and value of the project, identify requirements to address synergy and future dependency, assure compliance with architectural governance, and to support integration and alignment between projects.

This section describes development of architecture for one project within a portfolio. The effort starts with identifying the context, the superior architecture that defines the visions, the scope, and the value the project should deliver. Without initial exploration about where the project sits inside of the EA Landscape, Architecture to Support Project is in a volatile state. It is the responsibility of the Practitioners working in the Architecture Project to gather hints of uncovered barriers to the project. The project lies inside the roadmap at some linear point in a sequence of work packages. There are many hints from the roadmap alone of where to see danger ahead and who to ask about any unknown warning signs.

The purpose is to highlight the level of detail, time, and breadth during the ADM cycle phases for developing an EA as a focus of support to project architecture and governance. Most of the effort happens in the context of Phase F.

Table 8 summarizes the activities and use of appropriate steps from the ADM phases. The content of the table is discussed in detail in the rest of this section.

Table 8: Summary Table: ADM Phases and Architecture to Support Project

Topic	Mapping to TOGAF ADM Phase
Ascertain Dependencies	Partial Capability Level Phase A Program context: • Verify recency • Validate stakeholders, outcomes, timeline • Define project context in EA Landscape Partial Capability Level Phases B, C, and D Enterprise context: • Assess the readiness of the Enterprise to absorb proposed solution • Assess the ability of the solution provider to deliver the solution • Ascertain the scope of change of the Implementation Project Program context: • Elaborate architecture definition • Elaborate architecture specification • Define approach to minimize dependencies • Define risk controls

Topic	Mapping to TOGAF ADM Phase
Balance Options and Suppliers	Partial Capability Level Phases B, C, and D For each capability: - Define the solution boundary and conditions for integration - Validate continuity in EA Landscape - Create domain-specific work packages Partial Capability Level Phase E For each project in the portfolio: - Identify candidate ABBs and SBBs - Gather estimates - Develop project timelines - Trade-off impact with superior architecture - Update roadmap - Update risk matrix - Update work package and architecture specification
Finalize Scope and Budget	Partial Capability Level Phase F For each project in the portfolio: - Finalize estimates and timeline - Update Enterprise roadmap - Populate governance and approval plan Partial Capability Level Phase G For each project in the portfolio: - Perform architecture review - Confirm stakeholder approval Level Phase A For each project in the portfolio: - Candidate proof-of-concept work packages (as needed) - Candidate Statement of Work - Finalize stakeholder list - Draft success measures
Prepare for Solution Delivery Governance	Partial Program Level Phase F Program context: - Initiate completion of architecture work - Define target Solution Architectures - Finalize effort and resource estimates - Define variance measures in project-specific governance model - Update risk matrix

For Architecture to Support Project, the critical focus points are:

- Scoping:

  ○ What is the origin for the receipt of a Request for Architecture Work?

  ○ Where will I have overlap? Who are my neighbors (EA Landscape)?

- ○ Where do I look (EA Landscape: depth, breadth, detail)?

- ○ Are my stakeholders/portfolio guidance still relevant (recency)?

- Domain-specific stakeholders' concerns and architecture elaboration:

  - ○ Viewpoints/Stakeholder Map

  - ○ What do I need to know/solve for?

  - ○ Resolve impacts across architectures

- Finalizing the target transition architecture and its value:

  - ○ Creation of requirements and specifications

  - ○ Securing a Request for Architecture Work for the solution delivery architecture

Ascertain Dependencies

Throughout the entirety of the ADM, it is recommended to have a close look at the superior architecture in the EA Landscape. It is possible that Practitioners of superior architecture have already specified a list of things which the Practitioners of the preceding architectures are able to pull down to include as new inserts of an architecture description.

There is not much need to explore a reason to do architecture when the purpose of the project has already been specified within a roadmap. The Practitioner may find that they already have a sufficient Architecture Vision from the work that has been done in the portfolio architecture. However, the Practitioner must take responsible action to confirm the Architecture Vision along with a number of portfolio-level Target Architecture components to assess the impact of recency (see Introduction to the EA Landscape).

Assessing recency is the pulse of the Architecture Project. It will involve looking "bottom-up" at the current work in the EA Landscape to assess the impacts of recency to prior EA. Look at the set of Architecture Visions from the Architecture to Support Portfolio. The following set of questions will guide assessing the impacts of recency to prior EA work for the purpose of use:

- What EA is parallel in development?

- Which targets are in the process of being realized?

- Which targets have been approved?

- What is effect of recency on prior EA?

After prior EA work has been reviewed, reaffirmed, or replaced, the effect of recency is reset and mitigates the risk to the Architecture Project significantly.

Project is not a Magical Place to Swap Out Stakeholders

Who are the complete set of stakeholders across the architecture? The stakeholders in the portfolio level will need to be reaffirmed.

It is common to find organizational leaders, who at the starting of an Architecture Project, feel a strong need to replace the stakeholders identified in their superior architecture with stakeholders which have a high enough power to block or advance a *project* but not the architecture (see the TOGAF standard, Classify Stakeholder Positions, Section 24.3.2).

This will introduce new project-specific concerns into the architecture. It cannot be stressed enough, to hold on to the distinction between the stakeholders that have high power in the Architecture Project and those that have high power only in relation to the Implementation Project. At the end of the day, the Practitioner addresses the concerns of the empowered stakeholders holding the key to the success of the Architecture Project because they have the power to shape any Implementation Project in order to conform with the approved target. It may be useful to identify the project-specific stakeholders' concerns *if* we can solve for both and get something for free. Solving for an Implementation Project-specific concern is what can be called a "nice-to-have".

Stakeholders versus Key Players

Look at the previous Stakeholder Map from the portfolio. Assess recency. Map the complete set of stakeholders of the Architecture Project against their known concerns.

Do not include an Implementation Project-specific set of stakeholders (otherwise known as key players to the Architecture Project) in the Stakeholder Map. If desired, map the key players to any additional Implementation Project-specific concerns separately. Having more than one set of key stakeholders completely blocks the ability to perform trade-off.

Viewpoints and Requirements

The most important piece before doing any work; knowing what you need to know. Once you have a complete set of views which describe the stakeholders' concerns, you know exactly what you need to do, or at the very least, where to go look to find out what needs to be done.

When selecting viewpoints from the viewpoint library or developing new viewpoints, ask if the viewpoint represents the complete set of stakeholder concerns to the Architecture Project. Are all the stakeholders' classes representative of those which own the approval rights around the Target Architecture and decision rights around the Implementation Project?

Are the concerns consolidated and constrained into topic areas derived from the Enterprise's strategy, which will be consistent across Architecture Projects?

Does the viewpoint give a point of reference for what you need to know and where to look in the EA Landscape?

Once the Practitioner knows what information is needed and where to find it, it is safe to continue doing work without the fear of stepping on a figurative land mine.

Review the Architecture Repository for resources, especially architecture specifications, requirements, and work packages from the superior architectures to address the stakeholders' concerns for the Architecture Project.

Go Talk to the "Neighbors"

In developing a candidate architecture, the key to success is to be aware of the neighbors of the Architecture Project in the EA Landscape and to assess the "neighborhood" for recency. How much room is there for the Architecture Project until there would be an overlap or collide with another one?[30] When must you go and have a conversation with the neighborhood and assess their work for recency?

To add complexity, what is the current status of the neighboring architectures? Are the neighboring Architecture Projects approved, in transition, or becoming realized? You may not have to worry about rubbing shoulders with a neighboring Architecture Project until one of them enters a transition state.

Have the necessary conversations with the neighbors periodically to make the process of resolving impacts across Architecture Projects easier. The later these conversations with the "neighbors" takes place, the more likely the Practitioner will incur harder decisions, which would have been easily avoided; such as de-scoping decisions. The Practitioner must check the candidate architecture's flexibility to withstand the volatile environment shared with other Architecture Projects undergoing a number of transitional states.

Delivery and Acceptance Ability Assessment

This is an opportune time to assess the readiness of the organization to actually start to execute and realize the change. It involves identifying whether the work packages cover the necessary changes to business processes, operating procedure, training, and everything that has to happen once the solution is delivered. The assessment is narrowly focused to test the scaffolding the neighbors should have in place. A second set of assessment is the ability of the solution delivery team, internal or external, to deliver to the needs of the architecture specification. The project manager and the product owner are fully aware of the trade-off criteria to retain value; aware of dependencies from the neighbors to this effort and from this project to others; and the risks and controls to mitigate them.

Balance Options and Suppliers

Architecture to Support Project is to answer a set of problems in a box; the answers are expected to stay within the box. The Practitioner must elaborate all domain architectures just enough to assure that the architecture is addressing all of the work. The project cannot move forward until it is proven that the project will be a success. Gather the estimates of all resources required to deliver the project. All of the bridge will be built, not just some of the bridge. Remember, the focus is to clarify and confirm the purpose and value of the project. Part of the bridge does not serve any purpose or add any value.

The up-side is the Practitioners involved in the Implementation Project have blinders on that only allows them to see the distance from where they are standing to the horizon. The horizon is the work needed before implementation begins. In the context of the Implementation Project, the Practitioner's line of sight is always the horizon, including the distance to get there. It is already understood what "success" will look like, standing on the horizon. What is the work that will take us there?

---

[30] Is the Architecture Project in the Mojave Desert of the EA Landscape or in Abu Dhabi?

Performing Trade-Off

As the saying goes "you can't step in the same river twice"; the water's always changing, always flowing. Without discovery of where the candidate Target Architecture stands before finalization within the EA Landscape, it is harder to guide projects from running off waterfalls and large cliffs.

Only until the Practitioner looks "downstream" are they in a position to perform a trade-off, resolve impacts across the Target Architecture, and choose the smoothest course. Doing a consistent reconnaissance of the EA Landscape will enlighten the Practitioner to where the project can avoid disaster further down the river.

In order to perform, the Practitioner is chasing the barriers to deliver and realize value. This is too early to define the architecture for solution delivery. This is definitely not the place to define and design the solution. Implementation is not architecture. The architecture is assuring resilience to risks and guidance to implementers. Any recommendations of ABBs and SBBs to accelerate value realization and improve conformance are identified and included in an architecture specifications.

If it is discovered that the Implementation Project's candidate Target Architecture is impacting or will be impacted by a finalized Target Architecture of another project in-flight, always assess recency, confirm, and do a trade-off analysis. Keep in mind that when doing a trade-off analysis and resolving impacts across the Target Architecture that the Implementation Project is already heavily constrained and may need to mold a path down the river around the other projects that have been approved and have taken root along the river bank. Then, given any new discoveries to the Implementation Project, if any, create the architecture specifications for the Implementation Project to assure avoidance of overlap and conflict.

Managing the Current Approach towards Implementing the Change

Once impacts have been resolved, create the views necessary to convey to the stakeholders that their concerns have been addressed with the necessary constraints and guidance developed prior to initiation of solution delivery for it to be successful.

The Practitioner's analysis of the Target Architecture cannot have assessed every circumstance, or change option possible. There will always be an infinite number of things to discover about the Architecture Project. The Practitioner's job is to show that a sufficient level of scrutiny led to the deliverables of the Architecture Project for the solution delivery architecture to succeed. The Practitioner should only assess to the extent of avoiding major cliffs. Once you start assessing the Architecture Project for all the subtle bumps, you have exceeded the sufficient level of scrutiny and are wasting valuable resources.

Prove to the stakeholders that when the Architecture Project is consumed by the solution delivery architecture, their requirements have been met and changes to the Enterprise will be guided and constrained efficiently. Identify and secure approval for the resources necessary to begin allocating the budget for the solution delivery architecture to begin.

The Practitioner will know that the Architecture Project is a success upon receipt of the Request Architecture Work for solution delivery.

Finalize Scope and Budget

Implementation planning (Phase F) is the most critical piece in executing a walk through the ADM for the Implementation Project. Practitioners must rationalize for their Architecture Project what resources are required.

Package the project's architecture specifications, which includes the subsequent controls that mitigate the identified Implementation Project's risks. The package is then handed off to the Implementation Practitioner. It is the responsibility of the Architecture Project Practitioner to set up the Implementation Practitioner with everything they need to implement the project successfully.

If one or more work packages have not already been assigned to the Implementation Project, do so and seek approval. Be familiar with which gaps the work package(s) are filling and the purpose of their sequence in the roadmap. It may also be necessary to be familiar with the work packages the project will not be filling. Identify the risks within the work packages and subsequently within the Implementation Project.

Architect the "package" for the purpose of the Implementation Project. Create architecture specifications to the extent that an Architecture Project will not go off the rails on a crazy train. On the other hand, the railroads must not be easily scoured or constrained to the point of inflexibility of the volatile environment of the EA Landscape. Keep the Implementation Project on the tracks while maintaining the railways of the Architecture Project.

The Practitioner should package the architecture specifications including the principles, requirements, and controls within the context of the light shining down from the Architecture Vision of the portfolio, in the review of the Stakeholder Map, and the undertakings of the EA Landscape.

Refine the estimates and timeline for the project within the acceptable variance limits of the Enterprise. Cascade the update to project scope, trade-offs, and timelines to the Enterprise roadmap. Consult the requisite SMEs and stakeholders, and complete the architecture review. Populate the governance and approval plan for the solution delivery effort.

Prepare for Solution Delivery Governance

The maximum value is to be delivered by the Architecture Practitioner to the Enterprise in this step. Having finalized the scope and budget, make sure that the backlog information is complete for the project; trade-off, and decision criteria for the product owner, product manager, scrum master, or the project manager (whatever the role and title is) and the Implementation Practitioner is fully defined and understood; decision-makers and organizational leaders are fully aware of the barriers they must work to remove.

Any outstanding proof-of-concept work at this time should be limited to understanding an approach to the solution, not the architecture. Provide sufficient measurement criteria, indicators to warn of any variances, escalation, and deployment of SMEs, and implementation governance.

Initiate steps to close the Architecture Project. The Architecture Project's scope is limited to change management and governance. From that aspect, the project is not completed. This is also the time the architecture team and most of the Practitioners withdraw themselves from the limelight and pass the baton to Implementation Practitioners. Provide any required support for the Implementation Practitioners to defend the project during budget allocation. The work is not complete until the budget is allocated and the Implementation Project charter is signed.

Project Request for Architecture Work Originating from the Wild

The most common Requests for Architecture Work from the wild are for Architecture to Support Project. The central question for the Practitioner is to identify the proposed project's alignment to expected value and the

opportunity cost for the organization. See Request for Architecture Work Originating from a Random Idea from the Wild (on page 85) for a discussion.

Walk Through Architecture to Support Solution Delivery

Introduction

The success of this architecture and its outcome are driven by the degree of coordination between Architecture Practitioner and the Implementation Practitioner. The Architecture Practitioner hands over a well constrained, yet with sufficient room for creativity and innovation, box to the Implementation Practitioner. It is the duty of the Implementation Practitioner to not break the box or to morph its shape or appearance. It is the duty and responsibility of the Architecture Practitioner to define the context of this box within the EA Landscape, defining all of the push and pull forces. The candidate Architecture Project is now the Target Architecture.

Note that there will be minimal discussion on Phase G in Table 9. All of these activities occur in the context of Phase G. The table informs how activities in other phases enable delivery of the solution and drive closure to an Architecture Project. Actual closure is triggered from Phase H, either identifying a new effort or signaling achievement of target state.

Table 9 summarizes the activities and use of appropriate steps from the ADM phases. The content of the table is discussed in detail in the rest of this section.

Table 9: Summary Table: ADM Phases and Architecture to Support Solution Delivery

Topic	Mapping to TOGAF ADM Phase
Align Implementers	Partial Capability Level Phase A Project context: - Verify recency - Reaffirm stakeholders, outcomes, timeline - Communicate value proposition Partial Capability Level Phase B, C, D Program context: - Elaborate architecture specification - Reaffirm risk controls - Communicate SBBs Partial Project Level Phase G Program context: - Initiate project governance

Topic	Mapping to TOGAF ADM Phase
Guide Delivery	Partial Project Level Phases B, C, and D Project context: • Continuously update EA Landscape • Refine SBBs and solution boundaries • Monitor controls EA Capability specific context: • Update EA Repository (contents and models) • Update standards and reference architectures • Distribute resources Partial Capability Level Phase E Project context: • Analyze impact of trade-off with superior architecture • Update risk matrix Partial Capability Level Phase G Project context: • Conduct stakeholder review • Obtain architecture approval • Validate alignment of solution to vision
Realizing the Solution	Partial Project Level Phase H Program context: • Assess solution for gaps • Assess risk closure • Update Enterprise roadmap Partial Project Level Phase F Project context: • Baseline transition state architecture • Complete lessons learned • Close architecture work Partial Enterprise Level Phase H Program context: • Assess changes to Enterprise roadmap • As required, create backlog for architecture work EA Capability specific context: • Engage stakeholders • Update EA roadmap

Simple guidance for the Implementation Practitioner is to keep an eye on the target of the superior architecture. Be absolutely clear what the architecture is trying to optimize and what it is being asked to deliver. It may be tempting to remove all sub-optimization choices in the current delivery cycle. Refrain. Validate that sub-optimization is intentional and future work will address such concerns. All it takes is one bad driver to upset miles of traffic. Understand that the Solution Architecture is one of the many concurrently moving parts in the Enterprise.

Top concerns to be addressed in developing and delivering this architecture are covered in the following subsections.

Scoping

- What are the conditions under which a change can be triggered to architecture work?

- Having identified the neighbors and their interactions, what is the frequency of interaction and integration?

- What can and cannot give?

- Are the stakeholders and portfolio guidance still relevant (recency)?

Function Purity and Solution Innovation

- Are there multiple solution providers in this project? And who is providing what solution?

- What kind of detail is needed in the viewpoints to align solution providers and the superior architecture?

- How to drive integration across SBBs?

- How to select the best solution that aligns with the overall operating model (custom in-house, custom managed service, standardized managed service, standardized in-house)?

- What does governance mean in this context?

Handover and Closure

- When does the engagement end?

- What is the appropriate value report?

- What are the lessons learned and impact to gaps in EA?

Aligning Implementers

It is imperative that the Architecture Practitioner and Implementation Practitioner verify that the bottom-up view of the architecture aligns well on the "recency" measure. The next step is to validate the recency measure of the lateral set of architectures. The Architecture Project defines the boundary conditions to limit the impact to the overall architecture, accounting for all trade-off choices that would be made by the implementation architect. This doesn't mean that there cannot be changes to how each solution interacts with another. The impact does not require reprinting all of the training manuals and redoing the training schedule for the users of the solution.

In most cases, there would be more than one player; a solution provider and a solution consumer. The dynamic nature of business could ask for changes to the solution proposed mid-stream. The Architecture Project and hence the Solution Architecture clearly define the conditions that could trigger a change, stakeholder review, and architecture approval. A sizeable fraction of the projects will involve more than one solution implementer. Develop the architecture to identify, clarify, constrain, and liberate each of the solution

implementers from the other. The Solution Architecture articulates conditions for integration and acceptance of the total solution.

In-house or third-party solution implementers deliver against this architecture. When supplied by a third party, the onus is still on the in-house team to validate, integrate, and accept the solutions. At the end of the day, the consumers and end-users do not care who supplied the solution. Their question is: "Does this meet my expectations, does what it says, available as stated and defined?" Make sure that architecture, the governance plan and implementer are totally aligned on value proposition, conditions for trade-off, and the stakeholder matrix.

If the solution delivery project is validating a concept, the primary outcome is unearthing all points of failure; the secondary outcome is feasibility of the idea; and the tertiary outcome is scalability of the idea to meet usage demands. If the solution delivery project is building a bridge, its primary objective is enabling transportation under most environmental conditions; its secondary objective is to set terms of use. The variances across the solution delivery project are so vast that this paper cannot provide a sufficient set of examples to emphasize alignment with neighbors and completing the bottom-up view.

There is the least amount of work done in Phase A. It is all about affirming scope, stakeholders, currency, and value proposition.

Guiding Delivery

Any SBB delivered by solution suppliers will have to be integrated with the rest of the ecosystem of the Enterprise. Until the solution is delivered and evaluated against future work (transition architecture $n+1$), it will not be clear that some of the current work could become an SBB. Do not work to create a building block. Assess and refine once the solution is delivered and put to work.

In terms of architecture styles and patterns available in 2016, you may consider each Microservice or an aggregation of Microservices (SOA service) as an SBB.

When the superior architecture indicates availability of ABBs and SBBs, reach into the Enterprise Repository to reuse and conform to the architecture. When the ABBs point to implementations outside the Enterprise, guide industry collaboration and context-specific trade-off to guide development and delivery of Enterprise-specific SBBs.

Critical to success of architectures is retaining the ownership of integrating solution blocks within the Enterprise. Delegating the responsibility to any other party will lead to project management and governance issues, resulting in failed architecture.

Architecture to Support Solution Delivery is where all realizations and regulatory compliance needs are met. Naturally, the next critical long-term success factor for the Enterprise is identification of core information and data that should be retained in-house. The superior architecture should define the "core" for the Enterprise. All other datasets need not be retained, mastered, or controlled by the Enterprise. This choice drives other decision points in the operating model. Should the solution be treated as a black box for the Enterprise (a managed service) or specialized in-house or an expert team employed? Superior architectures need not resolve this choice. The choice and selection of solution provider is made at the time of developing and delivering the solution. Some of the solution provider choices may be constrained by the Enterprise's preference to restrict the number of suppliers. The Practitioner should not feel compelled to use a solution provider just because a constraint exists. Priority is fitness to deliver and accelerate time-to-market.

Choice of integration, definition of "core" information, and managed service *versus* in-house decisions guide the level of granularity needed to describe the architecture.

Populate the EA Landscape continuously; as each decision is made, the level of granularity of the architecture is arrived at, and interactions across solution blocks are defined. Quantifying and documenting the resource required by each solution block may not be the direct concern of the Implementation Practitioner or the Architecture Practitioner. Attributes like cost to procure, cost to deliver, and cost to operate are required by the Enterprise planning organization. It is a sensible option to capture these attributes within the EA tool. Financial investment data for each solution delivery project aids and reduces time to complete the trade-off analysis, roll-up and roll-down of budget, among other benefits.

It is not the recommendation of this paper that resource allocation data for solution delivery projects be mastered in the EA tool or the EA team to take responsibility. This paper is calling out a dataset that enables the Practitioner to be productive and purposeful. The source of truth for resource allocation should be determined by the Practitioner, following the guidance set by the Enterprise. A good content model and EA tool are normally capable of capturing this data point at the lowest level of granularity, and enable roll-up and trade-off analysis. It is the position of this paper to use an EA tool to do the computations that inform and impact trade-off analysis, instead of using other methods to speed up the time to inform trade-off.

Another set of trade-offs and constraints that impact this architecture is the existence of solution families in the Enterprise. The choice of a supplier or technology for data hosting services or ERP package constrains other building blocks that can be employed in the project and sometimes across the Enterprise. Take an assessment of such solution families from the superior architecture. When not available, the Implementation Practitioner and the Architecture Practitioner should spend time identifying, analyzing, and escalating impact of choices on large functional areas like Enterprise resource management and planning.

Even though the Architecture Project defines the boundary and the interface, change is bound to happen. Continuous interaction with the Architecture Practitioner and Implementation Practitioner is required to proactively mitigate barriers.

The objective is to develop the architecture to the extent needed to govern the solution being delivered. Do not feel compelled to define the solution as well. Define and employ viewpoints necessary to communicate, guide, and govern the Solution Architecture. Monitor implementation risks and the controls being implemented for Enterprise risks. Every trade-off and implementation choice made impacts and potentially modifies the Target Architecture. Governing the selections impacts the gap in the Target Architecture, the roadmap, and therefore the Architecture to Support Portfolio of the following fiscal year.

Work performed to deliver the solution mainly spans Phases B, C, D, and E. Innovations, research, and alternatives considered and employed follow the steps in Phase E. It is just that they do not go through rigorous architecture control. The alternatives are constrained by the architecture specification. Hence, it is a question of the ability to operate within constraints and not about controlling the selection. Specification created by following the steps in Phases B, C, and D assures appropriate selection.

Realizing the Solution

Contractually, this is the post-rollout, warranty period. Depending on the solution delivery method used in the Enterprise, this may be a parallel path to Guiding Delivery (on page 98). It is the period of putting the solution in the hands of the beneficiaries (customers, end-users, support personnel, partners, etc.). The

engagement of the Architecture Practitioner comes to a conclusion or shifts gear only when the solution is put to use. Depending on the appetite of the Enterprise, successful usage may be defined as the first 30, 60, or 90 days.

At the end of this period, the Architecture Practitioner initiates a gap analysis between the realized architecture and the Baseline Architecture to be used for solution delivery. It is only at the end of this analysis that a determination can be made about releasing key resources – the project manager, the implementation architect, supplier representative, technology resources reserved for developing the solution, etc. Closure of the Architecture Project is achieved as soon as the Implementation Practitioner accepts the superior architecture. However, the oversight provided by the Architecture Practitioner is retained until the solution delivery completion criteria are met.

Use the basis provided by the Architecture Project to report the value realized from time to time. Document the lessons learned, mainly the gaps in the description of the superior architecture that were filled while delivering the Solution Architecture. Document controls and constraints that accelerated overall delivery of the solution.

Update the cascading impact of the project to the EA Landscape and roadmap. As needed, validate, close and update the Enterprise backlog.

Project Request for Architecture Work Originating from the Wild

Requests for Architecture Work from the wild for Architecture to Support Solution Delivery are typically not done. Instead, there is a fully-baked Implementation Project with a proposed solution. In this case the Practitioner has to assess the fully-baked solution against the superior architecture. This becomes more of fitment analysis with its own political implication. See Request for Architecture Work Originating from a Random Idea from the Wild (on page 85) and Table 10: Example of Summary Governance Reporting (on page 123) for a broader discussion and assessment reporting example.

Conclusion

Most Architecture Practitioners fail in their role when supporting solution delivery. It is quite normal to confuse their role with SME, auditor, stakeholder, and proxy for the Enterprise stakeholder and decision-maker. Review Jumping to Phase G (on page 102) and Roles, Duties, and Decision Rights (on page 118).

The realized solution is the new baseline. It is the basis for evolving and analyzing the roadmap to the Target Architecture. All the development that happened in the Enterprise, and the industry, that were kept away from impacting solution delivery is added to the assessment set. This assessment is the next critical activity the Architecture Practitioner performs. It is this work that justifies closure of the current Architecture Project, Implementation Project, and resources. It also justifies the Request for Architecture Work for the next set of initiatives to achieve the target transition state ($n+1$). Involve all stakeholders, decision-makers, and implementers to complete the assessment, and gain the sign-off to close the effort.

Part 4: Guidance on Using an Enterprise Architecture

Jumping to Phase G

Most Practitioners will be regularly faced with their organization "Jumping to G". Many organizations select leadership on their ability to get things done. This creates a bias to action. Enabling effective change requires balancing predictable planned change with innovation and creativity.

Organizations that jump to Phase G will jump either because of organizational preference for visible action or execution failure by the EA team. In both cases, good Practitioners will respond to their organizational culture or to their failures. It is outside the scope of this paper for Practitioners to discuss effective engagement and Enterprise processes; see the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References).

The section will address classic failure patterns:

• Missing the purpose

• Missing the business cycle

• Not doing architecture

This section will also identify how the Practitioner addresses unpredictable change resulting from innovation, creativity, and circumstance.

An EA is developed for one very simple reason: to guide effective change. Guiding effective change involves serving decision-makers and implementers. Architecture to Support Strategy, Portfolio, and Project are focused on supporting decision-makers and are directly tied to planning stages in the business cycle. Architecture to Support Solution Delivery is primarily aimed at implementers. When the Practitioner does not provide timely support for strategy, portfolio, and project, the organization will continue to make decisions using the information at hand on the day the decision must be made.

Without a good Target Architecture to Support Strategy, Portfolio, and Project, the organization has jumped to Phase G. Typically this happens for two reasons: misalignment and missing the purpose.

Actual misalignment is outside the scope of this paper. For advice on the alignment of the EA, see the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References).

Most examples of misalignment in the industry are actually Practitioners missing purposes other than solution deployment.

Failure Pattern: Missing the Purpose

As clearly articulated earlier in this paper, different purposes require different architecture. The actual work product and analysis project to produce a view demonstrating to a change leader how a candidate architecture addresses agility for the purpose of strategy is radically different than for the purpose of solution deployment. Practitioners must adapt the basic structure and concepts to different purposes. Too much advice masks the essential differences by using terms such as high-level or aspirational or conceptual or logical. A good Practitioner will know how to distinguish high-level work for the purpose of strategy from high-level work for the purpose of solution delivery.

Every stakeholder and every concern are addressed in every purpose.

Practitioners miss the purpose when they tell themselves stories about breadth, depth, and timeframe. As discussed in Introduction to the EA Landscape, there is a set of rough guidelines regarding breadth, level of detail, and planning horizon. Further, regardless of the exact parts of the EA Landscape that must be addressed by any particular architecture development project, a Practitioner will find themselves without clean edges.

Architecture to support a purpose is typically aligned to support different points in the business cycle, and required to inform different decisions, as all work must be aligned to the purpose at hand. This may change the key work product's essential purpose, but is unlikely to substantially change which components in the architecture must be analyzed.

Failure Pattern: Missing the Business Cycle

Most leaders are interested in receiving effective advice about complex decisions. Usually, the Practitioners are waiting for an invitation to a planning process that will never come. Leaders may be surrounded by parochial champions who wish to pitch their pet projects. In response, they actively seek to reduce involvement in planning processes to those who provide useful, balanced advice and those they wish to hold accountable for the change.

Delivering architecture to support the business cycle requires being ahead of decisions. The Practitioner works ahead of the planning cycle (see Figure 4 on page 35). For many Practitioners, working ahead of the planning cycle is an uncomfortable position. They must be focused on preparing for activities that no one else is thinking about.

For example, Architecture to Support Portfolio facilitates the budget process for an organization that operates an annual budget process. With such a cycle, the budget finalization is likely done near the end of the third quarter. This requires the budget planning to be done near the end of the second quarter, which requires the first draft of the candidate Target Architecture and candidate roadmap to be available for the second quarter. Stakeholders and decision-makers are then able to use the candidate architecture and candidate roadmap in planning and preparing their budget submission and defending their submission in any resulting budget negotiations. The Practitioner then needs to understand their candidate material is used, stretched, and changed through the entire budget preparation and negotiation. In short, the Practitioner is involved in iterating through Phase E and F through the second and third quarter.

Practitioners who are unfamiliar with the give-and-take typical in most organizations' planning processes will wait for clarity or decision. Both are only available at the end of the planning process, not in the middle. As a result, the Practitioner has missed their place in the business cycle.

Architecture after Decision

This paper is designed to assist Practitioners to deliver useful architecture. Architecture produced after decision is not only useless but dangerous. At best, the architecture will validate the decision. Given the decision has already been made by leaders with the authority to make the decision, validation is pointless. At worst, the architecture will demonstrate the leaders made the wrong decision. It is technically useful to gain this knowledge and perform a course correction. The damage to the EA team and wasted time and effort executing the next steps following the decision are unlikely to be compensated by a better decision.

Practitioners adept at establishing value will be keenly aware of the impact time has on almost every value calculation. Lastly, Practitioners adept at estimating the cost of change will be keenly aware of how expensive misfires are on the ability of an organization to execute an effective change.

Few activities a Practitioner can perform are as dangerous as architecting after decision.

Failure Pattern: Not Doing Architecture

Practitioners will often fulfill multiple roles in the architecture development and change process. Architecture Governance (on page 117) identifies stakeholder, SME, architect, implementer, and auditor as the essential roles in architecture development. Practitioners will typically act as an agent for the stakeholder, making decisions by proxy through their understanding of the set of stakeholders' preferences. Many Practitioners, by way of their growth path, would have expert knowledge in specific domains; they will tend to provide advice and guidance as SMEs to stakeholders, other architects, and implementers. Some Enterprise's structure may demand a Practitioners to act as implementer. An implementer normally pays attention to details like product selection, configuration challenges, assuring quality and repeatability, etc. These tasks are often sufficiently time-consuming that the Practitioner does not have time to perform architecture.

Many EA teams fall into the trap of performing implicit architecture. The Practitioner is so busy acting as a stakeholder's agent, SME, and implementer that the architecture is never described and approved by a stakeholder. A work product that is really implementation design, and implementation specification and standards definition is provided as the end result of the "Architecture Project". These work products are the end result – they are not architecture.

Architecture Governance (on page 117) will discuss the need to deeply review implementation work products that exist unsupported by architecture description, views, and architecture specification. Bluntly, what evidence can a Practitioner provide that the implementation is in conformance with the architecture, provides the best available approach to addressing the stakeholders' preferences and the organization's mission, vision, value proposition, and objectives? The only choice is compliance by assertion.

Compliance by assertion is rife with personal bias and "tourist dashboard decisions".

Practitioners deliver value not by tripping over the correct implementation but by facilitating the complete set of stakeholders to understand the implications of their preferences in the context of the Enterprise's mission, vision, value propositions, and objectives. Whether this is done on the easy path by preparing views addressing concerns or by facilitating trade-off between competing decisions is immaterial. The absence of understanding means the architecture, and the value it enables, is fragile. The moment the Practitioner is unengaged on landscape, there can be no expectation that the value will be sustained by operational teams and future implementation teams who are unaware of either preference, priority, or traceability to value.

Without an architecture, the Enterprise has no choice but to jump to Phase G – completely unprepared, with no ability to exercise implementation governance.

Not performing architecture to support decision-makers and implementers is the most pernicious practice a Practitioner can perform.

Managing Innovation, Creativity, and Circumstance

Top-down direction and planning provides part of the answer for a nimble organization. It provides the guidelines, constraints, and clarity required to make tactical decisions. Sometimes the correct decision is to embark on unplanned change.

Whether the Practitioner has arrived at implementation of change unprepared because of a failure or because of a good deliberate decision, the Practitioner still needs to provide useful support of the change activity. Stakeholders simply have to have less confidence that the project will deliver the expected value with the expected cost and the projected time. The range of unknown ones precludes high confidence.

This lack of confidence simply means the architecture has more uncertainty, or risk, associated with realizing the organization's objectives. At this point, Practitioners have to focus all of their energy on risk mitigation.

Pragmatically the Practitioner is going to be constantly performing a risk management function. Rather than diving into the details of implementation the Practitioner needs to find and expose uncertainty associated with the objective to provide tactical governance support. Every project will have some form of benefits statement. Every organization has some form of strategy. The Practitioner simply has to connect the dots without the benefit of any intermediate stepping stones. The important distinction here is that the Practitioner is not expected to correct the project regarding benefits statement and realization plan. The Practitioner is expected to mitigate uncertainty regarding realizing the benefits stated in the project.

TOGAF Phase G provides a step for this activity where the Practitioner provides guidance to the Implementation Project. The Practitioner must walk a line between guiding and performing implementation.[31] Implementers are expected to live within the constraints of the project; Practitioners are expected to look at the context of the project. The most valuable actions when the organization jumps to Phase G are identical to addressing rapid implementation methods such as agile. The Practitioner must focus on the scope of the Implementation Project, facilitating good decision-making in the context not of project benefits realization but of Enterprise benefits realization, and ensuring the stakeholders and implementers understand the implications of their choices regarding Enterprise benefits not driving them to make different choices. This is a very fine distinction and is it a reiteration of not fixing the project but ensuring stakeholders and implementation teams understand what can honestly be expected in terms of value and benefit.

Innovation and creativity are at the fore when an organization jumps to Phase G. Thoughtful architecture development providing guidance and constraints at the required level of detail will be missing. When the Practitioner's organization is in a hurry they are focused on receiving value through differentiation and experimentation. Typically, a sustained efficiency gain is not achieved without clarifying dependency. Practitioners should expect that organizations in a hurry are usually fully aware of the difficulty sustaining experiments across time and when scaled. Hence, the Practitioner must focus on value realization. Bluntly, this is not different than a more thoughtful approach: The stakeholders' preference and priority drives the architecture development.

---

[31] For a discussion of the different roles a Practitioner may play, see Failure Pattern: Not Doing Architecture (on page 103), and Roles, Duties, and Decision Rights (on page 117).

In terms of the TOGAF ADM phases, the Practitioner will be running constant micro-iterations exploring discrete statements of value through to the implementation, with the purpose of clarifying the value expected and what in the implementation creates uncertainty. In order to perform this, the Practitioner will have to focus all attention on a narrow set of concerns on the critical path to value realization.

When the organization Jumps to Phase G, the Practitioner will routinely need to act as the stakeholders' agent. Practitioners must be keenly aware of the danger acting as both the architect and the stakeholders' agent. Care must be taken to guard against tunnel vision, personal bias, and "tourist dashboard decisions". Specialized reporting against the narrow set of concerns on the critical path to value and the Implementation Project form the control that mitigates lack of preparation and failing to separate duties.

Special Cases

Architecture in an Agile Enterprise

There has been a great deal of conversation about aligning to agile implementation methods. Ink has been spilled trying to align the phases of the ADM to these development methods. All of this conversation has blurred the line between implementation and architecture. The TOGAF standard aligns to agile development in Phase G. Full stop.

A good Architecture to Support Portfolio, or Project, will identify what products the Enterprise needs, the boundary of the products, and what constraints a product owner has. In short, a good architecture defines the Enterprise's backlog.

Architecture to Support Project and Solution Delivery will have a set of constraints that limit the choices of the agile team. These constraints are where an individual product must bend to Enterprise issues and the parochial preference of a product owner is not valid.

Then Phase G, Implementation Governance: the Practitioner serves the stakeholders guarding the mission, vision, goals, and investment roadmap. In short, guarding Enterprise value.

Architecture for a Domain

A common failure path is for domain architects to work to a different purpose, or pretend that they are working on a different Architecture Project than the rest of the team. A domain[32] must fit into the whole of the EA. Also, the rest of the EA must fit with a domain. Anything else is a tourist dashboard decision (see Tourist Dashboard Decisions on page 66).

A distinct domain is security. A security architecture only exists in reference to other domains and is best considered a concern. Practitioners will always address their stakeholders' security[33] and risk concerns.

Architecture in Response to an Incident

Something happened, and the organization's response is to fix it.

As a starting point the Practitioner should understand risk as the effect of uncertainty on reaching objectives, risk appetite, and risk tolerance. Achieving all objectives is uncertain, and an Enterprise's response is driven by risk tolerance and risk appetite.

---

[32] See Appendix H: Evolving List of Domain Architectures (on page 139).

[33] For a detailed discussion, read the referenced Open Group Security Forum Guide: Integrating Risk and Security within a TOGAF® Enterprise Architecture.

The risk appetite provides guidance balancing the amount of risk taken to achieve an expected outcome. Risk appetite is typically expressed as a boundary on a risk/business impact and likelihood grid, or qualitative measures. For example, the Enterprise will risk $x for $y reward this year, or has zero tolerance for loss of life. A well understood risk appetite defines both the level of risk the organization is willing to accept as well as its strategy in defining this level. For risks above this acceptable level, it defines the strategy used for mitigation. Strategy for risk in excess risk appetite is typically transference or avoidance.

Risk tolerance addresses deviations from what is expected. In short, what to do when the Enterprise's uncertainty is exceeded. The most common expression of uncertainty is failure to achieve expectations. At this point, the Enterprise is certain it will not achieve its objectives.

An incident changes the stakeholders' preferences with regard to risk. This is a change in requirement, and the architecture must adjust. The central role of the Practitioner is to provide solid advice on what changes to the target, and the associated work to achieve the change will reach an acceptable certainty of reaching the stakeholders' objective. Practitioners should not be surprised when there are few changes that have an acceptable cost, and the stakeholder is faced with the option of cancelling the objective or cancelling the change.

Part 5: Guidance on Maintaining an Enterprise Architecture

Transition Architecture: Managing Complex Roadmaps

Until now, this paper made the effort and process simple by describing most of the concepts using a linear time scale. It gave an impression that creating a well aligned set of work packages vectored by business cycle and planning horizon gives you potential transition states and a near linear roadmap. Recall this simple statement made in Coordination Across the EA Landscape and EA Team, in the context of the EA Repository: "Baseline provides reference for all change. The target state is what stakeholders have approved. Transition states are partially realized targets between current state and target state. Mix the four characteristics of the EA Landscape: breadth, depth, time, and recency. Mix the different Architecture Projects that can work on the same subject at different times and at different levels of detail." That's the only hint to indicate real-world complexity.

In addition to characteristics, other organizational factors that add to complexity are:

- Advancements and changes outside the Enterprise

- Shared services

- Collaboration with suppliers and partners, including portfolio ownership model

- Impenetrable dependencies

- Multiple geopolitical boundaries (fiscal calendars, regulations, cultures)

- Varying rate of maturity and growth of teams

- EA team model (federated, centralized, etc.)

- Availability of multiple solutions or announcement of end-of-life for products currently in use

This is the realty. One Enterprise roadmap gets broken down into segment, portfolio, or geography. The Enterprise will be pursuing more than one concurrent goal, say efficiency and retooling. For each business cycle, the roadmap is revisited to make adjustments, bottom-up and at times top-down. This is a clear use-case that drives the need for a good EA Repository: a repository that maintains the integrity of the current state and target state, but allows creation of variants.

Roadmap Grouping

Start with one version that supports the initial strategy. Flesh out the repository from strategy to project. Upon acceptance of the portfolio, create versions as necessary. Once the candidate versions are accepted, baseline both current and Target Architectures. Create multiple baselines of the current transitional state. Create copies of the architecture, one per variable, concern, or a related group of variables.

Use the same planning horizon to showcase the impact and outcome. The moment planning horizons change, analysis becomes complex and results in loss of continuity for most decisions.

Each distinct parent roadmap – say if there is a separate roadmap for European Union Operations and Australian Operations – name and identify them as such. Employ appropriate naming and versioning

concepts for and derived roadmaps of those created for what-if analysis. Make it intuitive to identify discarded alternatives.

Comparing Architectures

The point of creating separate roadmaps is to align the scope of each Architecture Project. When the Enterprise has any one of the characteristic or organizational factors identified earlier in this section, it would make sense to create a separate Architecture Project and roadmap to deal with this complexity.

Employing a standard reference architecture for process, business terms, applications, etc., supports cross-project and cross-roadmap analysis. Using a standard model provides the flexibility required to map across implementation models of the solution suppliers. It also helps in evaluating bids and offers from potential suppliers. This is another place where use of ABBs would come in handy. Implementation and use of ABBs across projects can be analyzed with ease.

Basing all of the architectures on an implementation-neutral reference model allows impact of modifications to a specific architecture to be identified easily. As shown in Figure 17, the EA Repository tool can provide visual clues to identify the change, whether it is to one of the attributes of an architectural component or a modification to the catalog of components. While working with a federated team, the tool and use of common reference models goes a long way to coordinate and communicate the impact of architecture changes. Within the roadmap, it is better to keep the analysis patterns consistent.
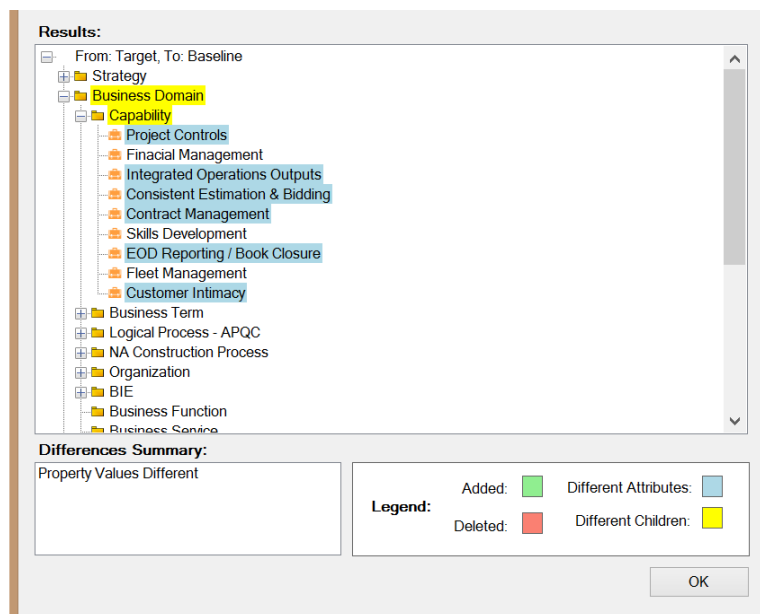

Figure 17: Using Repository for Managing Roadmaps – I

This same concept of comparing architectures can be used to create and analyze year-over-year modifications to the architecture. In Figure 18, notice that the EA Repository tool allows the Practitioner to trace a change to the baseline or the revised version.
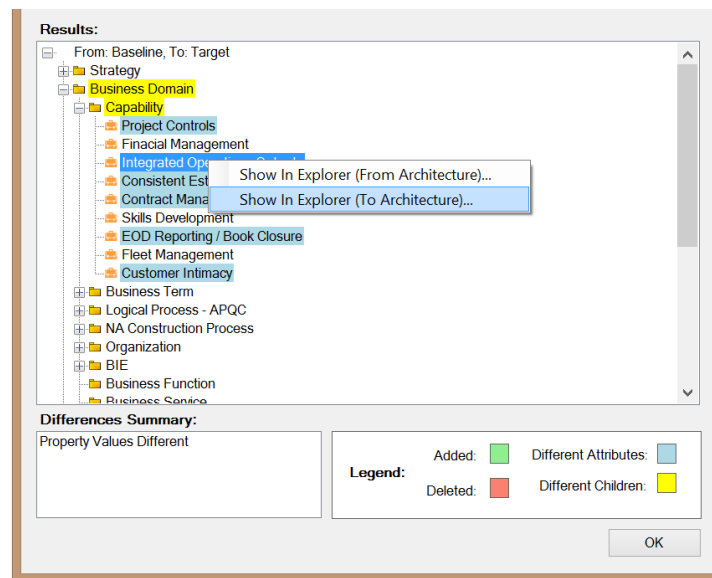
Figure 18: Impact Analysis of Architectures

When creating the roadmap, pay attention to impact of change. Any change, when introduced, will tarnish the efficiency, overall throughput, and sometimes call for duplicative investments. Such short-term negative impacts can mask deviations from the roadmap. Inject appropriate markers to identify any unintended sub-optimization or deviations from the roadmap. The value and outcome map should present the time to value and gain/loss at the end of the planning horizon.

General Guidance

A work package or an architecture specification that intersects more than one Architecture Project or change effort also introduces complexity. The environment for every Enterprise is highly dynamic, forcing a need for trade-off and expert judgment every so often. Implementation Projects are invariably insulated from all impact from developments in the external environment. Complexity happens because every transitional state is a fully functional and operational state for the Enterprise. The architecture and roadmap evolve to stay abreast or ahead of such external changes.

When starting afresh, the Practitioner potentially has the benefit of working with the limited set of information about the landscape. As the landscape is populated from ongoing Architecture Projects, continually pay attention to ruthless abstraction of detail. Set your biases and baggage aside. Set the stakeholder preference aside. It is all about the least and absolute necessary information to guide a choice. Keep the dataset consistent. Eliminate noise and distortions when performing analysis of architectures.

Common traps while creating roadmaps include incorrect scoping. The Architecture Project may exclude certain functions from the scope. Earlier sections of this paper explicitly warned you not to stray away from the charter of the Architecture Project. The fine-print is that, if you identify a need, a gap, call it out – don't work on developing the architecture. It is the responsibility of the Practitioner to call out the dependency and document its existence and the disposition of the gap in the roadmap. Such deferred items will become its own roadmap. When developing architecture for this gap at a later date, make sure that you operate in a fixed

block of time (same end dates as related roadmaps), not a fixed block of duration (say three years for each roadmap).

Phase H (Coordination and Business Cycle in Action)

An EA is developed for one very simple reason: to guide effective change. The change can be materialized only when it is adequately supported with resources. Every Enterprise has a business cycle that plans and allocates resources, normally one fiscal year. The fiscal year dates are inflexible and decisions will be made with the data available and reasonable judgment.

If the EA Capability has been requested by the Enterprise, it is an acknowledgement of the fact that "implicit" architecture and the resulting judgments that drove investments and changes are not delivering what the Enterprise wants. It is likely that the EA effort was kicked off after the budget allocation for the current business cycle or with very limited time to influence the decisions of the current business cycle. Do not waste time in the current cycle. Stay happy with the "implicit acknowledgement" and focus on building the data for the next cycle. Though not stated, the sponsor is looking to protect "future" decisions with EA. The moment the Practitioner realizes they are late for the next cycle, shift the time investment to refurbish the résumé of the entire team (see Failure Pattern: Missing the Business Cycle on page 103).

Phase H demands the Practitioner to identify the bottom-up drivers for change; change due to improvements in available technologies or conditions controlling the operations or environment of the Enterprise; and initiate the architecture work for the next target transition state (top-down driver). This does not mean that the Practitioner need to flesh out everything that is covered by the charter for the EA Capability or the budget.

Earlier sections impressed upon "just enough architecture" and characteristics of the EA Landscape. Understand the capacity and capability of the team to scope the work. Remember, the definition of "Enterprise" is fungible and used to control the scope of analysis. If this is the first pass in developing the Architecture to Support Strategy, scope the effort accordingly. Define and distribute the work packages in proportion to the capability and readiness of the Enterprise. All of these are aimed at one thing – influencing and garnering the resources in the next cycle.

Tying everything to the budget cycle simply highlights the importance of good EA in guiding and constraining the change decisions. When there is no practical input from a good EA team before the decision an organization needs to take is made, the decision is still made. It might even be a good choice, but it was a less informed choice.

The moment there is awareness that data was available, but late, irrespective of the quality of the decision made, the EA team loses its relevance. It is a fail-fail scenario resulting in questioning the value and purpose of the EA team.

Depending on the size of the Enterprise (irrespective of the scope of the EA work), budget preparation may start two to four months before the start of the fiscal year. The Practitioner, the Implementation Project architect, and the Implementation Project manager need to play the role of SME to assess the ability of the implementation team to complete all the work packages at least two to four weeks before the start of budget preparation date.

Other than the first year of operation of the EA team, most subsequent architecture work is initiated from Phase H. Phase H provides ongoing review of value realization and monitoring of change. Change and failure to realize value provide entry points to the ADM. Never be late – four weeks before the start of budget planning is too late. The EA team needs to be aligned with the organization's planning, budgeting,

operational, and change processes. Figure 19 shows a timeline view, depicting an alignment of key decisions made during a business cycle and the purpose architectures.
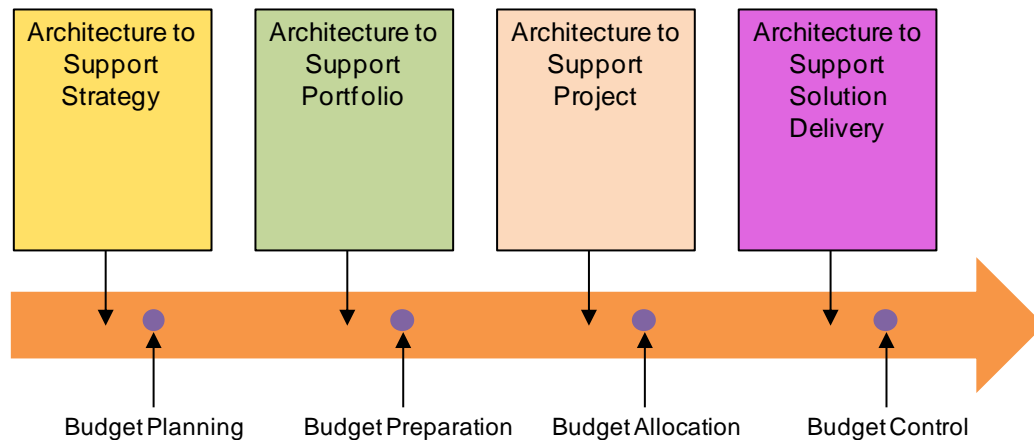

Figure 19: Business Cycle and Architecture by Purpose

Once the Practitioner's communication informs and influences the budget planning, the path forward is set. This superior architecture governs and constrains the rest of the activities.

The second most important activity is supporting budget control. The architect of the Architecture Project is the agent for the stakeholder for the implementation team; the architect is also the SME for the portfolio manager in validating the progress earned to value. It is common to see a Practitioner tripped by the duality of role in the budget control phase to lose focus on the budget planning activities. Never forget that the sole purpose of the Practitioner is to influence and guide change – not to get into the detail of implementation.

The EA team is intentional about every effort, irrespective of the name used – process improvement, operations, Keep-The-Lights-On (KTLO), growth, transformation, etc. Every effort and idea contributes to the Target Architecture. Even through the superior architecture constrains the Architecture to Support Portfolio and Project, nothing is committed and accepted as the next transition state until resources (budget) are allocated. Random ideas from the wild (see Request for Architecture Work Originating from a Random Idea from the Wild, on page 85) will find their way into the process. The Practitioner watches like a hawk to identify such interesting work packages and triggers a review, trade-off, and governance of the "new" portfolio. Unless sufficient insight is gained about the "behavioral" patterns of the organization, it is difficult to discern "pet projects" and "random ideas" disguised as "bottom-up" effort from a legitimate initiative to bridge a gap. Perform a simple sniff test – is the architecture specification trying to accomplish more than one thing; stakeholder trade-off – are the concerns aligned or being accepted for lack of time to analyze. Create a change request and leave a bread crumb to revisit and stabilize the architecture in the next cycle.

Understand how the Enterprise employs discretionary funds; use them wisely. A practical approach would be guiding allocation of such discretionary funds for exploratory work packages, until the alignment to roadmap could be rationalized and included in the portfolio. Acceptance of such requests is an explicit change to the Target Architecture. Avoid as much as possible. Follow the change management processes. No exceptions. The role being played by the Practitioner at this stage is more of a mediator and negotiator, applying the architectural knowledge. At the end of the day, the Practitioner is responsible and accountable for the stability and integrity of the architecture.

At the time of finalizing the allocation of funds, good architecture will speak for itself. The Practitioner need not be in the room to guide the decision. When the allocation happens, the decision-makers are validating that the project manager, portfolio manager, and the implementation architect fully understand and agree to deliver the outcome in conformance to the architecture. The decision-makers are already convinced of the need for the project and its outcome. If the Practitioner enters a scenario requiring change to the architecture, it is too late. The foundation is faulty. The Architecture Project and the Implementation Project cannot proceed. Go back to the architecture specifications and stakeholder concerns. Be prepared to face the consequence of incomplete work.

If the Practitioner had followed everything in this document up until Special Cases (on page 107), everything mentioned in this section should appear to be a foreign concept. Otherwise, start over with this document.

Architecture Governance

ISO/IEC 38500:2015[34] defines governance as: "a system that directs and controls the current and future state". The process by which direction and control are provided should imbibe equality of concern and transparency, protecting the rights and interests of the organization.

Governance is a decision-making process, with a defined structure of relationships to direct and control the Enterprise in order to achieve stated goals. The key difference between governance and management rests on the cornerstone of fiduciary and sustainable responsibility.

Most discussion on governance confuses management and governance. John Carver's Policy Governance is written to support public agencies, where there are often competing priorities and strong distinctions between those who pay and those who benefit. It is one of the best pieces of guidance a Practitioner can get. Lastly, John's work clearly distinguishes between governance and management. The parallels to EA governance are striking.

The development and use of EA must be governed. To define a customized governance approach, let us start to define the following:

- What is to be governed?

- Why should something be governed?

- When and who should decide on the recommended alternatives?

What is Governed and Why?

Two distinct things must be governed. First, the development of the Target Architecture. Second, all change within the scope of the Target Architecture. Without the first, the Practitioner cannot support their organization's leadership directing and controlling change. Without the latter, there was no point in developing a good target that provides an organization's best achievable course forward.

Central to the definition of governance is "directs and controls". Typically, the Practitioner and implementer are directed, and both are controlled by the stakeholder. This section will use the terms direct and control for focus.

Target Architecture

The TOGAF standard provides a key concept to govern the Target Architecture: the Architecture Project.

The Architecture Project is used to direct and control the EA team to address issues in the Enterprise. An Architecture Project starts with a Request for Architecture Work. The primary control is Architecture Project

---

[34] ISO/IEC 38500:2015: Information Technology – Governance of IT for the Organization.

management using the Statement of Architecture Work. For a broader discussion of controlling the development of the Target Architecture, see the Architecture Project Management White Paper.[35]

In short, the Practitioner is directed to develop an architecture within a controlled scope. Within that controlled scope, the Practitioner is directed to the stakeholder's preferences. Preferences are expressed in terms of objective, priority, and specification. Best practice requirements management chases objective and priority as the baseline. The governance test will ask whether the Practitioner is addressing the stakeholder's concerns.

Implementation Projects and Other Change

The TOGAF standard provides two key concepts to govern Implementation Projects and other change: the Architecture Contract and the Architecture Requirements Specification.

The Architecture Contract is used to direct and control the implementation team to work towards a deliberant future. Regardless of the document structure an Architecture Contract takes in a Practitioner's organization it will contain the same directional elements and provide a means to test compliance.

The Architecture Requirements Specification is used to direct and control the creativity of the implementation team. Every Architecture Requirements Specification enables control of the implementation team. Design, implementation, and other change choices can be tested against the Architecture Requirements Specification.

In short, the implementation team is directed to create changes with intentional value-based outcomes. Best practice governance enables the organization to control value realization.

Roles, Duties, and Decision Rights

Decision rights about the Target Architecture, relief, and enforcement are always vested in the architecture's stakeholders. The most common failure pattern is to confuse roles.

Each role is involved in the governance of developing and using architecture, with different accountability and decision rights. The roles are:

- **Stakeholder**: Owner of the architecture. Provides priority, preference, and direction. All decision rights about the Target Architecture, and any relief from and enforcement of the target, are vested in the stakeholders.

- **Stakeholder Agent**: Representative of the stakeholder.

- **Subject Matter Expert**: Possesses specialized knowledge about some aspect of the Enterprise or the environment in which it operates. Provides knowledge, advice, and validation of interpretation.

---

[35] Architecture Project Management: How to Manage an Architecture Project using the TOGAF® Framework and Mainstream Project Management Methods (see References).

- **Implementer**: Responsible for performing all change activity. Scope of change is not relevant. Transformative capital projects and incremental operational changes are changes performed by an implementer. All decision rights about proposed implementation choices, such as design, product selection, and change sequence, are vested with the implementer.

- **Architect**: Developer of the Target Architecture. Provides recommendations when non-compliance with the target is determined.

- **Auditor**: Performs systematic reviews of both the target and implementation. Best performed at multiple stages to capture errors before the cost of correction exceeds potential value realization. All decision rights about compliance during the development of the architecture and implementation are vested with the implementer. Auditing can be performed within a formal structure such as an architecture governing board or by a peer reviewer. Auditing can also be self-performed but the role being performed needs to be clear in the mind of the individual and that they are acting in accordance with the role.

In many organizations, the Practitioner will fill the role of stakeholder agent, subject matter expert, and implementer. This typically occurs when the organization does not use architecture to direct and control change. Instead, the organization attempts to use skilled thoughtful individuals to make tactical decisions. The value is illusionary.

The governance process does not have to be a heavyweight bureaucracy. It is simply based on demonstrating sufficient traceability that the organization can have confidence in the target being the best path to reaching the Enterprise's preferences. With confidence, the Enterprise will enforce the target in deliberate change activity.

Target Checklist

Use the following checklist to execute architecture governance. Good Practitioners understand that only stakeholders can approve architecture. A good governance process will require the Practitioner to demonstrate the following when assessing a Target Architecture:

1.	Were the correct stakeholders idenfied?	Yes/No
		If yes, proceed.
		If no, direct the architect to engage with the stakeholders appropriate to the scope of the architecture being developed.
2.	Were constraints and guidance from the superior architecture taken into account?	Yes/No
		If yes, proceed.
		If no, direct the Practitioner to perform their job and take into account guidance and constraints from the superior architecture. Where the Practitioner identifies a conflict, obtain a recommendation on whether to grant relief from the superior architecture or enforce the superior architecture. This decision must be made by the superior architecture stakeholders.

3.	Do appropriate SMEs agree with the facts and interpretation of the facts in the architecture?	Yes/No
		If yes, proceed.
		If no, the Practitioner has to do their job and engage with the SMEs. Where the Practitioner identifies a conflict with, or between, SMEs, develop a recommendation for the stakeholders that they should have limitations in confidence.
4.	Do any constraints or guidance produced reflect the views produced for stakeholders and any underpinning architecture models and analysis?	Yes/No
		If yes, proceed.
		If no, the Practitioner needs to do their job and develop appropriate views that are consistent with analysis.
5.	Do the views produced for the stakeholders reflect their concerns and reflect any underpinning architecture models and analysis?	Yes/No
		If yes, proceed.
		If no, the Practitioner needs to do their job and develop appropriate views.
6.	Do the stakeholders understand the value, and any uncertainty in achieving the value, provided by reaching the target state?	Yes/No
		If yes, proceed.
		If no, the Practitioner needs to do their job and develop appropriate views, and other work products, then return to the stakeholders.
7.	Do the stakeholders understand the work necessary to reach the target state and any uncertainty (risk) in successfully accomplishing the work?	Yes/No
		If yes, proceed.
		If no, the Practitioner needs to do their job and develop appropriate work products and return to the stakeholders.
8.	Do the stakeholders understand any limitations in confidence they should have in the Target Architecture?	Yes/No
		If yes, proceed.
		If no, the Practitioner needs to do their job and develop appropriate guidance on the limitations in confidence and return to the stakeholders.
9.	Have the stakeholders approved the views?	Yes/No

If the answer to the last question is yes, the governance process is done. The architecture, associated view, architecture specifications, controls, and work packages are ready for publication in the EA Repository as an approved Target Architecture.

If the answer to the last question is no, then there is a decision on whether the Practitioner should rework the architecture or the Architecture Project should be cancelled. Reworking the architecture typically requires the Practitioner to finally embrace the stakeholder's preferences. Rework may require more advanced trade-off.

Implementation and Other Change Checklist

When the architecture is being used, changes to the Enterprise are guided and constrained. Two factors impact governance of change. First, organizations operate in a dynamic environment, and the analysis of the Target Architecture cannot have assessed every circumstance or change option possible. Second, the target was produced for a purpose and may not have been developed to the level of detail required for the current use. The governance process requires flexibility. When non-compliance is identified, the Enterprise must either change the architecture, provide temporary relief from constraint, or enforce the architecture. If relief is not temporary, the Enterprise has chosen the worst available option: changing the target without bothering with analysis and approval.

Two governance roles are often performed by the Practitioner: the auditor and the architect. Compliance assessment is an auditor role. When non-compliance is identified, the architect needs to produce an impact assessment and recommendation on what to do. The recommendation will have three choices: First, enforce compliance; second, provide temporary relief; and third change the Target Architecture.

The choice in the recommendation will be driven by the impact assessment. Practitioners must assess impact on the same terms as the target was developed. Assessing on any other terms invalidates the assessment and recommendation.

Implementation governance assesses compliance. Compliance assessment needs to be done soon enough that course correction is viable. As identified in the walk-through sections, compliance assessment against value and operational change are as important as project-driven change.

This checklist is designed to assist the Practitioner understand what must be demonstrated during the governance process to address a non-compliance report:

1.	Did the organization embarking on a change reasonably interpret the Target Architecture's guidance and constraints?	Yes/No
		If yes, their interpretation should be accepted as compliance and any issues addressed through a change to the architecture. This is a key point. Good architecture can have multiple implementation choices, and the implementer is not required to adhere to opinion. If the implementation choice is a reasonable interpretation, it should be judged compliant.
		If no, proceed.
2.	Do appropriate SMEs agree with the facts and interpretation of the facts in the impact assessment?	Yes/No
		If yes, proceed.
		If no, the Practitioner has to do their job an engage with the SMEs. Where the Practitioner identifies a conflict with, or between, SMEs, develop a report for the stakeholders identifying what limitations in confidence they should have in the impact assessment.
3.	Do appropriate SMEs agree with the recommendation to enforce the target, grant time-bound relief, or change the	Yes/No
		If yes, proceed.

	architecture?	If no, the Practitioner has to do their job and engage with the SMEs. Where the Practitioner identifies a conflict with, or between, SMEs, develop a report identifying what limitations in confidence the stakeholder should have in the compliance recommendation.
4.	Do the views and other materials produced for the stakeholders reflect the impact assessment and reflect any underpinning architecture models and analysis?	Yes/No
		If yes, proceed to the stakeholders for approval.
		If no, the Practitioner has to do their job.
5.	Do the stakeholders understand any limitations in confidence they should have in the impact assessment?	Yes/No
		If yes, proceed.
		If no, the Practitioner has to do their job and provide the appropriate work products that highlight the impact of limitations in confidence and return to the stakeholders.
6.	Do the stakeholders understand the impact on prior expected value, and any change in certainty in achieving the value, provided by reaching the target state?	Yes/No
		If yes, proceed.
		If no, the Practitioner has to do their job and provide the appropriate work products that highlight the impact on expected value, and on uncertainly in reaching the expected value and return to the stakeholders.
7.	Have the stakeholders approved the recommendation to enforce the target, grant relief, or change the architecture?	Yes/No

If the answer to the last questions is yes, the organization should action the recommendation. How this is actioned is context and organization-specific. Where compliance is enforced, the governance process should look for evidence of a course correction to the Implementation Project. Lastly, where relief is provided, the Practitioner should ensure that future compliance assessment and reporting take place to review time-bound relief. Without this step, the Enterprise has simply agreed to change the Target Architecture without the bother of approval.

If the answer is no, the stakeholder has spoken. A Practitioner can make the choice to try and convince the stakeholder through expanded information provided to the stakeholder. One of the common mistakes is that the Practitioner either switched terms of assessment from those used to develop the target, or failed to embrace the stakeholder's preferences when developing the impact assessment.

Long-Term Compliance Reporting

The sections discussing walk-throughs for Architecture to Support Strategy, Portfolio, and Project all included assessments of in-flight change and consider using summary reporting with a high visual impact. Below is an example of reporting against constraints, expected value, and known gaps. In all cases, the assessment will return either not applicable, conformance, or non-conformance. Good Practitioners will look for binary tests: compliance and con-compliance (Red/Green) where possible. Where binary testing is not possible, a 1-to-3 scale (Red/Yellow/Green) should provide sufficient range to provide a summary report.

Table 10: Example of Summary Governance Reporting

	Constraint (Architecture Principle, Architecture Requirements Specification, or Control)	Value (Best done in terms of the Enterprise's mandatory concerns)	Gap
Current state: assess what the Enterprise has	Conforms	Fails to Deliver	Not Applicable
Implementation Project: assess project, design, and implementation	Violates	Not Applicable	Filling
Roadmap, portfolio, or program: assess plans and directions	Not Applicable	Delivers	Leaving Open

Conclusion

The Practitioner serves the Enterprise's stakeholders regardless of where they are employed in an organization. This requires the Practitioner to identify with and guard the stakeholders' preferences. Good Practitioners use their position in front of decisions and outside of the change program to guard value. In practice, a high fraction of governance is informal, with the Practitioner thinking as the stakeholders' agent and deciding when to push for compliance. For every change initiative, understanding and guarding the Enterprise's expected value is the most important and arguably the only job of architecture governance.

Part 6: Appendices

Appendix A: Partial List of EA Content Frameworks

Table 11 provides a list of alternative EA Content Frameworks. Specific mapping papers exist between the TOGAF standard and BIAN, DoDAF, Frameworx, and SABSA (see References).

Table 11: Partial List of EA Content Frameworks

Framework	Framework Description
AGATE	The France DGA Architecture Framework
BIAN	Banking Industry Architecture Network
Deloitte EAF	Deloitte Consulting Enterprise Architecture Framework
DNDAF	The Department of National Defence Architecture Framework (Canada)
DoDAF	The US Department of Defense Architecture Framework
FDIC-EAF	FDIC Enterprise Architecture Framework (US)
FEAF	Federal Enterprise Architecture Framework (US)
Frameworx	TM Forum
GEA	Government Enterprise Architecture – Queensland Government
MoDAF	The UK Ministry of Defense Architecture Framework
NAF	The NATO Architecture Framework
Navigate	Conexiam Enterprise Architecture Content Framework
NIST EA	NIST Enterprise Architecture framework (US)
NORA	Nederlandse Overheid Referentie Architectuur (The Netherlands)
OBASHI	The OBASHI Business & IT Methodology and Framework
OEAF	Oracle Enterprise Architecture Framework
PEAF	Pragmatic Enterprise Architecture Framework
PERA	Purdue Enterprise Reference Architecture Framework
SABSA	The SABSA Institute Enterprise Security Architecture
TEAF	Treasury Enterprise Architecture Framework (US)
UAF	United Architecture Framework (replacement for UPDM)
UPDM	United Profile for DoDAF and MoDAF
Zachman	Zachman Framework

Appendix B: Partial List of Reference Models and Reference Architectures

Table 12 provides a list of reference models and reference architectures. These examples are provided as a starting point.

Table 12: Useful Reference Models and Reference Architectures

Reference Model & Reference Architecture	Use
APQC	American Productivity and Quality Center process reference available in cross-industry and industry-specific models
ARTS Information Model	Association of Retail Technology's information and data model
BIAN	Banking Industry Architecture Network
Frameworx	TM Forum
The IT4IT Standard	The Open Group Business, Application, and Information reference model for the IT function
Navigate Information & Application Atlases	Conexiam Enterprise Architecture Content Framework
PPDM	Professional Petroleum Data Management upstream petroleum industry information and data model
SCOR	Supply Chain Operations Reference (model)

Appendix C: Partial List of Modeling Approaches

Table 13 provides a list of modeling approaches. These examples are provided as a starting point for a Practitioner who needs to consistently describe some part of an Enterprise.

The EA community is filled with involved discussions of the distinction between language, notation, model kind, and model type. Such fine-grained distinctions are normally not useful. What is useful is describing something consistently.

These approaches may have a formal or informal metamodel, notation, or supporting method.

Table 13: List of Useful Modeling Methods

Reference Model & Reference Architecture	Use
4+1 architectural view model[36]	Can be used in Architecture to Support Solution Delivery. The four views of the model are logical, development, process, physical view, and use-case. Provides a nice simplified list of what you need to know and describe.
The ArchiMate Standard	Excellent fit for Architecture to Support Solution Delivery. Good fit for Architecture to Support Project.
Business Model Canvas[37]	Use is entirely driven by the scope of the value proposition. Commonly used for Architecture to Support Portfolio and Architecture to Support Project.
Business Motivation Model (BMM)[38]	Simplified is useful for Architecture to Support Project. Can be used for Architecture to Support Portfolio BMCs.
Business Process Model and Notation (BPMN)[39]	Can be used for Architecture to Support Solution Delivery. Limited fit for analysis required in architecture.
Kaplan Strategy Map[40]	Good for representing final strategy.
Organigraphic	Very useful in looking at a governance model of an Enterprise. Use is driven by the scope being described. Commonly used for Architecture to Support Portfolio and Architecture to Support Project.

[36] Refer to Kruchten: Architectural Blueprints – The "4+1" View Model of Software Architecture (see References).

[37] See: http://businessmodelgeneration.com/canvas/bmc.

[38] See www.omg.org/spec/BMM/Current/.

[39] See www.omg.org/spec/BPMN/2.0/.

[40] Refer to Kaplan and David: The Balanced Scorecard (see References).

Reference Model & Reference Architecture	Use
A3 Thinking[41]	Useful in summarizing Architecture to Support Project.
Unified Modeling Language (UML)[42]	Good fit for Architecture to Support Solution Delivery. In particular, useful in providing a standard way to visualize the design of a system.

[41] See http://sloanreview.mit.edu/article/toyotas-secret-the-a3-report/.
[42] See www.omg.org/spec/UML/2.5.

Appendix D: Stakeholder/Concern Matrix

In Conexiam's Navigate, a set of standardized classes of stakeholders, concerns, and associated viewpoints are maintained for each architecture purpose. This follows the advice of aligning the EA Capability with the questions that are expected to be answered.[43] This section provides a partial list of common stakeholders, concerns, and their alignment. These examples are provided as a starting point for a Practitioner who needs to address common questions.

Table 14 shows the relationships between the stakeholder classes and concerns for a single architecture purpose.

Common Stakeholder Classes

- **Senior Leaders** are those with responsibility for management and oversight. This responsibility includes approving and realigning strategic initiatives, tracking a portfolio of projects, ensuring transformative benefits are realized, and meeting operational business goals.

- **Program/Portfolio Managers** are those with responsibility for management and oversight of strategic initiatives. This responsibility includes approving and realigning projects, tracking project progress, and ensuring project benefits are realized.

- **Business Requirements Owners** are those responsible for identifying and expressing business requirements. Typically, these stakeholders are responsible for some aspect of business operation.

- **Implementers** are those responsible for developing, integrating, and deploying the solution.

- **Risk Owners** are those interested in risk.

- **Business Partners** are those who are engaged to provide services sustaining a customer value proposition.
  Note: The architecture may not be provided to business partners, but must be evaluated from their perspective.

- **Customers** are those who consume products and services.
  Note: The architecture may not be provided to members, but must be evaluated from their perspective.

Common Concern Classes

- **Agility**: what is the ability of the architecture to adapt to future unanticipated change?

- **Efficiency**: how does some aspect of the architecture contribute to efficiency of operations?

- **Differentiation:** how does some aspect of the architecture address enable differentiation?

---

[43] See Customization of Architecture Contents and Metamodel in the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References).

- **Value**: what is the value of the architecture?

- **Value Proposition**: how does some aspect of the architecture address a value proposition?

- **Change Cost**: what is the impact of a change to the architecture in terms of cost of change?

- **Change Impact**: what is the impact, or scope, of a change to the architecture?

- **Alignment**: to what extent is the architecture aligned with priorities?

- **Feasibility**: what is the probability the architecture will be realized and sustained?

- **Dependability**: how will the architecture consistently deliver value and operate safely?

- **Control:** how will we protect assets in the architecture?

- **Specification**: what needs to be built?

- **Security**: will the architecture consistently address the risks and opportunities embedded in operations?

- **Confidence**: what confidence can be placed in the target?

- **Customer Intimacy**: is the Enterprise delivering products and services the customers want? What is the confidence that the new product or service will be liked by them?

- **Scalability**: Can the architecture and the Enterprise handle the range of demands and growth cycles?

- **Business Continuity**: Does the architecture provide the appropriate level of continuity needs relative to the Enterprise's needs?

For each intersection, a viewpoint is created the identifies the necessary information and communication required to address the concern. (See Appendix E: Sample Viewpoint Library, on page 132.)

Table 14: Stakeholder Responsibility (Portfolio)

	Agility	Efficiency	Value	Value Proposition	Change Cost	Change Impact	Alignment	Feasibility	Dependability	Control	Specification	Security	Confidence	Customer Intimacy	Scalability	Business Continuity
Senior Leaders	X	X		X		X	X					X		X		X
Portfolio Managers	X	X		X		X	X	X					X	X	X	X
Business Requirements Owners	X	X		X		X					X	X	X	X		
Implementers						X		X		X	X		X		X	
Risk Owners						X		X	X		X	X	X		X	
Business Partner	X	X				X		X			X	X	X			
Customer	X			X								X	X	X		X

Appendix E: Sample Viewpoint Library

In Conexiam's Navigate, a Viewpoint Library is maintained to identify the standard concerns, stakeholders, and the information required to address the question. The information is typically drawn from one or more models. How the view should be constructed is purpose-specific.

Table 15 shows the relationship between the stakeholder classes and concerns:

Table 15: Viewpoint Library (Portfolio)

Concern	Stakeholders	View Construction	Information Required
Agility			
Efficiency			
Value			
Value Proposition			
Change Cost			
Change Impact			
Alignment			
Feasibility			
Dependability			
Control			
Specification			
Security			
Confidence			

Appendix F: Solution Delivery Notebook

In Conexiam's Navigate, a Solution Delivery Notebook is maintained to standardize communication from an architecture to a solution delivery team.

Table 16 shows the relationship between the stakeholder classes and concerns.

Table 16: Solution Delivery Notebook

Section	Part	Purpose
Solution Summary		This section provides the summary of the solution. Central is: • What set of gaps in the architecture does the solution address? • Who are the stakeholders, relevant inbound requirements, and relevant specifications that address the requirements? • What are the risks, and the relevant controls that address the risks?
	Solution Concept Diagram	Describes the central problem and how the solution addresses the problem.
	Stakeholder Catalog	Identifies key stakeholders, their requirements, and any associated architecture specifications that constrain the design and implementation. This allows any design and implementation to be tested against stakeholder requirements by tracing the design and implementation to the requirement through the architecture specification.
	Risk Catalog	Identifies the risks applicable to the solution and the mitigating controls. This allows the design and implementation to be tested against risk though the mitigating control.
	Gap Catalog	Lists gaps that are addressed by the work package. This identifies what is in scope of the project and what is not in scope. Keep in mind there will routinely be additional gaps that are not addressed by a project that will need to be identified as such.

Section	Part	Purpose
Specification Summary		This section provides the summary for testing the design and implementation against the architecture and provides the basis of architecture governance. Specification conformance will be tested against: • Requirement/specification pair • Risk/control pair
	Implementation Strategy	Identifies the preferred approach to addressing the gaps or work packages, where a preferred approach will improve value realization.
	Architecture Specification	Identifies all the specifications that address a requirement. Specifications can be of many different types. Note that the specification can apply to anything in the architecture, but always traces to a requirement.
	Control	Identifies all the controls that mitigate a risk. Note that the control can apply to anything in the architecture, but always traces to a risk.
Architecture Description Summary		This section provides the summary of the Target Architecture using appropriate diagrams, catalogs, and matrices. This section is provided for reference.
	Business Architecture	
	Information Architecture	
	Application Architecture	
	Infrastructure Architecture	
	Security Architecture	
	Other specialized domain architecture	

Appendix G: Another ADM Journey: Leader's Guide Capability-Based Planning Journey

This White Paper has focused on aligning use of the TOGAF standard to support four primary purposes driving the development of an EA. The journeys described in Walk Through Architecture to Support Strategy (on page 70), Walk Through Architecture to Support Portfolio (on page 78), Walk Through Architecture to Support Project (on page 87), and Walk Through Architecture to Support Solution Delivery (on page 95) provide purpose-specific journeys.

Practitioners will face many journeys through the ADM.

Table 17 is from the TOGAF® Leader's Guide to Establishing and Evolving an EA Capability (see References). It outlines a customized journey through the TOGAF ADM that is optimized for an EA Capability; it is easily adapted to other capability-based planning Architecture Projects.

As always, Practitioners identify the information they need to know to answer the question at hand. These answers either inform the next question and/or support a decision. Effective iteration of the ADM is not linear.

Table 17: Mapping EA Capability Development with ADM Phases

Topic	Mapping to TOGAF ADM Phase
Enterprise Context and EA Context	Partial Strategic Level Phase B Enterprise context: • Goals, objectives, initiatives, competitive, and tactic analysis • Operating model (partners, suppliers) • Explore what-if scenarios and scorecards EA context specific for the EA Capability: • Goals
Business Objectives for the EA Capability	Capability Level Phase A For the EA Capability: • Provide initial goals and objectives • Select a reference EA Capability and maturity model • Candidate EA Capability • Candidate operating model • EA Capability gap and priority roadmap

Topic	Mapping to TOGAF ADM Phase
Architecture Governance	Partial Segment/Capability Level Phase B For the Enterprise: • Enterprise Risk Management Model • Governance Model For the EA Capability: • Risk Management Model • Governance Model • Extend candidate operating model to include EA governance • Initial Architecture Partition Model • Trace to EA Capability goals
Alignment with Other Frameworks	Partial Capability Level Phase B & Partial Phase C (Data) For the Enterprise: • Reference models for key frameworks • Capability assessment of key frameworks For the EA Capability: • Framework touch-points • Extend candidate operating model to include other frameworks • Extend EA governance and EA risk management • Initial EA Content Framework aligned to other frameworks and EA governance • Candidate architecture partition model • Trace to EA Capability goals • EA Capability and key framework gap and priority roadmap
Customization of Architecture Contents and Metamodel	Capability Level Phase C (Data) For the EA Capability: • EA Content Framework • EA Content Metamodel • Viewpoint Library • Architecture Repository Model • Trace to EA Capability goals • Initial EA Content Framework and Architecture Repository gap
Organization Model for the EA Team	Partial Capability Level Phase B For the EA Capability: • EA organizational model • Select reference EA skills framework • Initial alignment with Enterprise job titles and roles • Initial accountability matrix for EA Content Framework and initial Architecture Repository • Organizational gap and priority roadmap

Topic	Mapping to TOGAF ADM Phase
Process Model	Partial Capability Level Phase B Capability Level Phase C (App) and Capability Level Phase D For the Enterprise: • Process model highlighting touch-points between EA Capability and Enterprise processes the EA Capability supports[44] • Performance matrix for key processes and organization • Accountability matrix for EA Content Framework and organization For the EA Capability: • Process model • Architecture Repository application model • Matrix for EA Content Framework and Architecture Repository Applications Architecture • Process and Architecture Repository gap and priority roadmap
Create the EA Capability	Capability Level Phase E Create a roadmap highlighting development of the EA Capability by changes in the: • Organizational model • Process model • EA Content Framework • Architecture Repository For the EA Capability: • Trace roadmap to EA Capability goals
Establishing and Evolving the EA Capability	Capability Level Phase F and Capability Level Phase G For the Enterprise: • Transition the EA Capability Roadmap to an Implementation & Migration Plan For the EA Capability: • Execute the Implementation & Migration Plan to build the EA Capability the Enterprise desires

---

[44] While this has been stressed in the paper, align to processes the EA Capability is expected to support based upon its purpose. Do not align to those it could support. Worst practice is to fret over linkage to processes the EA Capability *could* support.

Appendix H: Evolving List of Domain Architectures

As the ecosystem in which an Enterprise operates and information technology evolves, specialty domain architectures will evolve. Table 18 documents a partial list of domain architectures and a short note about the domain. The list or the note about the domain should not be considered authoritative or comprehensive.

Table 18: Partial List of Domain Architectures

Domain Architecture		Short Note about the Domain Architecture
Business Architecture		Focuses on business motivations and business operations, linking customers, products, services, finances, suppliers, and partners. The linkages, relationships, and operational aspects are elaborated using the Enterprise's goals, objectives, strategies, business processes, and capabilities along with its rules and controls.
Security Architecture		An approach that clearly addresses the necessities and potential risks involved in a certain scenario or environment. It also specifies when and where to apply controls to eliminate or mitigate the barriers to attain the objectives, including sustainability and continuity of business.
Service Architecture		An approach to describe the purpose and method of interaction to get an outcome for the buyer/user. Includes clear articulation of the service availability, location, access control, response expectations, and usage methods.
	Human Machine Interaction Architecture	An approach to study and optimize the effort and understanding required by humans to work with machines and applications.
Information Systems Architecture		This is a logical grouping describing processes that are automated. The description includes information accessed and produced, infrastructure used to host applications that automates the processes, communicates across applications, or stores information. This is composed of all information, data, application, infrastructure, communications, and integration architectures.
	Information Architecture	A structural design and approach to help users (humans and machines) understand where data (text, audio, video, binaries) is, how to find it, what to expect, and how to use it to improve quality of decisions.
	Data Architecture	A description of policies, rules, or standards that govern which data is collected, how it is stored, arranged, integrated, and put to use. Organization of data is normally expressed in models.
	Application Architecture	Describes the behavior of a solution (automated or manual) applied to solve a business problem, how the solution interacts with other such solutions, and its users. It also describes how the solutions are organized, including its structural and behavioral elements.

Domain Architecture		Short Note about the Domain Architecture
	Infrastructure Architecture	A description of elements without which core business operations cannot take place. In generic terms, includes buildings and space for parking, power supply, heating, ventilation and air conditioning systems, dining area and restrooms (in other words facilities). In the information technology context, covers bare metal computing devices like servers, routers, switches, and disks.
	Communications Architecture	A network of people and machines that connects separate components of an organization. The primary focus of this architecture is to enable flow of information across the organization and rest of the world. Normally includes telephony, video conferencing, and automated response systems.
	Integration Architecture	A description of tools and techniques applied to enable applications to interact with each other using appropriate communications and infrastructure architecture. Its focus is on setting rules of engagement between applications including protocols and method, compliant with risk and security architecture.

References

The following documents are referenced directly in various sections of this paper or the concepts discussed in this paper are derived from these works.

(Please note that the links below are good at the time of writing but cannot be guaranteed for the future.)

- ArchiMate® 3.0 Specification, an Open Group Standard (C162), June 2016, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/c162.htm.

- Architecture Project Management: How to Manage an Architecture Project using the TOGAF® Framework and Mainstream Project Management Methods, White Paper (W16B), August 2016, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w16b.htm.

- John Carver: Reinventing your Board: A Step-by-Step White Paper to Implementing Policy Governance, Jossey-Bass, 2006.

- Jeff Conklin: Wicked Problems & Social Complexity within Dialog Mapping: Building Shared Understanding of Wicked Problems, Wiley, 2005.

- Donald C. Hambrick, James W. Fredrickson: Are you Sure you have a Strategy?, The Academy of Management Executive, 15, 4; ABI/INFORM Global, November 2001.

- Exploring Synergies between TOGAF® and Frameworx, White Paper (W114), May 2011, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w114.htm.

- Integrating Risk and Security within a TOGAF® Enterprise Architecture, Open Group Guide (G152), January 2016, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/g152.htm.

- Integrating the TOGAF® Standard with the BIAN Service Landscape, White Paper (W135), October 2013, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w135.htm.

- ISO/IEC 38500:2015: Information Technology – Governance of IT for the Organization.

- ISO/IEC/IEEE 42010:2011: Systems and Software Engineering – Architecture Description.

- Robert S. Kaplan, David P. Norton: The Balanced Scorecard – Measures that Drive Performance, Harvard Business Review, 70(1), Jan-Feb 1992.

- Philippe Kruchten: Architectural Blueprints – The "4+1" View Model of Software Architecture, November 1995; refer to: www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf.

- Henry Mintzberg, Bruce Ahlstrand, Joseph Lampel: Strategy Bites Back: It is Far More, and Less, than You Ever Imagined, April 2005.

- TOGAF® 9 and DoDAF 2.0, White Paper (W105), July 2010, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w105.htm.

- TOGAF® and SABSA® Integration, White Paper (W117), October 2011, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w117.htm.

- TOGAF® Version 9.1, an Open Group Standard (G116), December 2011, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/g116.htm.

- World-Class Enterprise Architecture, White Paper (W102), April 2010, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w102.htm.

- The TOGAF® Leader's Guide to Establishing and Evolving an EA Capability, Open Group Guide (G168), December 2016, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/g168.htm.

Suggested Reading

- Cuypers Ataya: Enterprise Value: Governance of IT Investments, The Business Case, IT Governance Institute, 2006.

- Peter Swartz: The Art of the Long View: Planning for the Future in an Uncertain World, Currency Doubleday, 1996.

- Kees van der Heijden: Scenarios: The Art of Strategic Conversation, 2nd Edition, Wiley, 2005.

About the Authors

Dave Hornford, Conexiam

Dave Hornford is Conexiam's Managing Partner and leads Conexiam's Boston practice. He is the owner of Conexiam's Navigate and Navigate Atlases. Dave serves on the board of trustees of The SABSA Institute. He is the former Chair of The Open Group Architecture Forum and was a key contributor to the TOGAF® 9 standard. Based in North America, he works in a variety of industries including financial services, oil and gas, technology, and capital-intensive industry. Typically, he helps clients develop and execute a roadmap to transform.

Nathan Hornford, Conexiam

Nathan Hornford is a management consultant and ABACUS Certified Architect and Designer. Nathan is based in Canada. Nathan owns Conexiam's Navigate Content Framework enabling predictable Enterprise Architecture, specializes in formal modeling, and supports all architecture analysis, option analysis, and roadmap development. Nathan works with all of Conexiam's practices to provide consistent architecture methods and tools that address the client's change needs.

Sriram Sabesan, Conexiam

Sriram Sabesan is an Open Group Certified Distinguished Architect. Based in North America, he specializes in technology, manufacturing, telecommunication, and financial services industries. Sriram helps clients to develop and execute strategies in response to digital or economic disruptions. He is actively involved in development of different Open Group standards.

Sadie Scotch, Conexiam

Sadie Scotch is an Enterprise Architect. Sadie is based in the US and is a member of Conexiam's Boston practice. Sadie specializes in governance, option analysis, and roadmap development. She helps clients to develop and govern change programs to address current Enterprise priorities.

Ken Street, Conexiam

Ken Street is an Enterprise Architect. Based in Canada, he leads Conexiam's Governance and IT4IT™ initiatives. He is the current Vice-Chair of The Open Group Big Data project and is active within The Open Group IT4IT™ and Open Platform 3.0™ Forums. He works primarily in financial services and oil and gas, helping clients to develop their EA Capability, improve their IT organization, and execute architecture-driven change programs.

Samantha Toder, Conexiam

Samantha Toder is a management consultant and ABACUS Certified Architect and Designer. Sam is based in the US. Sam owns Conexiam's Pilot guides for executing predictable Enterprise Architecture, and is actively involved in shaping the next version of the TOGAF® standard. She helps clients to develop in-house EA Capability and execute complex transformation programs in the financial services industry.

Acknowledgements

About The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 500 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – to:

- Capture, understand, and address current and emerging requirements, establish policies, and share best practices

- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies

- Offer a comprehensive set of services to enhance the operational efficiency of consortia

- Operate the industry's premier certification service

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