

THE WHOLE FRAMEWORK IN PLAIN LANGUAGE



TOGAF, Simply Explained

Learn it. Say it simply. Find the gap.
Explain it again.

One story • Nine ADM phases • Four domains • Zero unnecessary jargon



An independent learning guide aligned with the TOGAF Standard, 10th Edition

What is inside

1. How to use the Feynman Technique
2. What TOGAF is—and is not
3. The whole architecture journey
4. Every ADM phase in simple words
5. The four BDAT domains
6. Core terms and architecture content
7. Reuse, repositories, and governance
8. Practical techniques and modern delivery
9. A complete Sunny Toys example
10. Exam thinking and the SCOPE method
11. A one-page memory sheet and tiny glossary
12. Fifteen questions that prove understanding

How to use this book

Richard Feynman was famous for explaining hard ideas in simple words. His learning method is easy:

1. **Learn one small idea.**
2. **Close the page.**
3. **Explain it as if you are teaching a child.**
4. **Notice where your explanation becomes fuzzy.**
5. **Study that small gap, then explain it again.**

This book follows that pattern. Each section has:

- a plain explanation;
- one example;
- a memory hook; and
- a **say it back** challenge.

Do not try to memorize every sentence. Build the picture in your head.

Your finish line: You can explain TOGAF without using the word “TOGAF” in the explanation.

Our one example: Sunny Toys

Sunny Toys has ten shops. It wants customers to:

- see which toys are available;
- buy online;
- collect from any shop; and
- receive the same good service everywhere.

Today, every shop works differently. Stock numbers are often wrong. There are three customer lists and five old applications. Sunny Toys needs coordinated change—not just a new website.

We will use Sunny Toys for the whole book.

FEYNMAN CHECK

Say it back: What are the five steps in the Feynman learning loop? Explain them without looking.

1. What TOGAF is

Imagine a town wants to build homes, roads, schools, pipes, and parks.

If every builder works alone:

- one road may end at a wall;
- pipes may not reach the homes;
- two schools may be built where only one is needed; and
- the town may spend money without becoming a better place to live.

The town needs a shared plan. It also needs a way to create, check, and update that plan.

An enterprise has the same problem. Its “roads and buildings” are:

- people and teams;
- business capabilities and processes;
- information;
- applications;
- technology; and
- projects and products changing all of them.

Enterprise Architecture is the shared picture and decision discipline that keeps those parts moving toward the same outcomes.

TOGAF is a framework for doing Enterprise Architecture. It gives you:

- a repeatable method;
- a shared vocabulary;
- ways to describe architecture;
- analysis techniques;
- governance practices; and
- advice for building an architecture capability.

Shortest correct explanation: TOGAF helps an organization understand where it is, decide where it wants to go, plan the changes, and keep delivery moving in that direction.

What TOGAF is not

TOGAF is not:

- a finished architecture you can copy;
- a software product;
- a diagramming language;
- a project management method;
- a command to create every possible document; or
- a rigid waterfall.

It is a **method and toolbox**. You choose the parts that help with the real decision.

Three words that sound bigger than they are

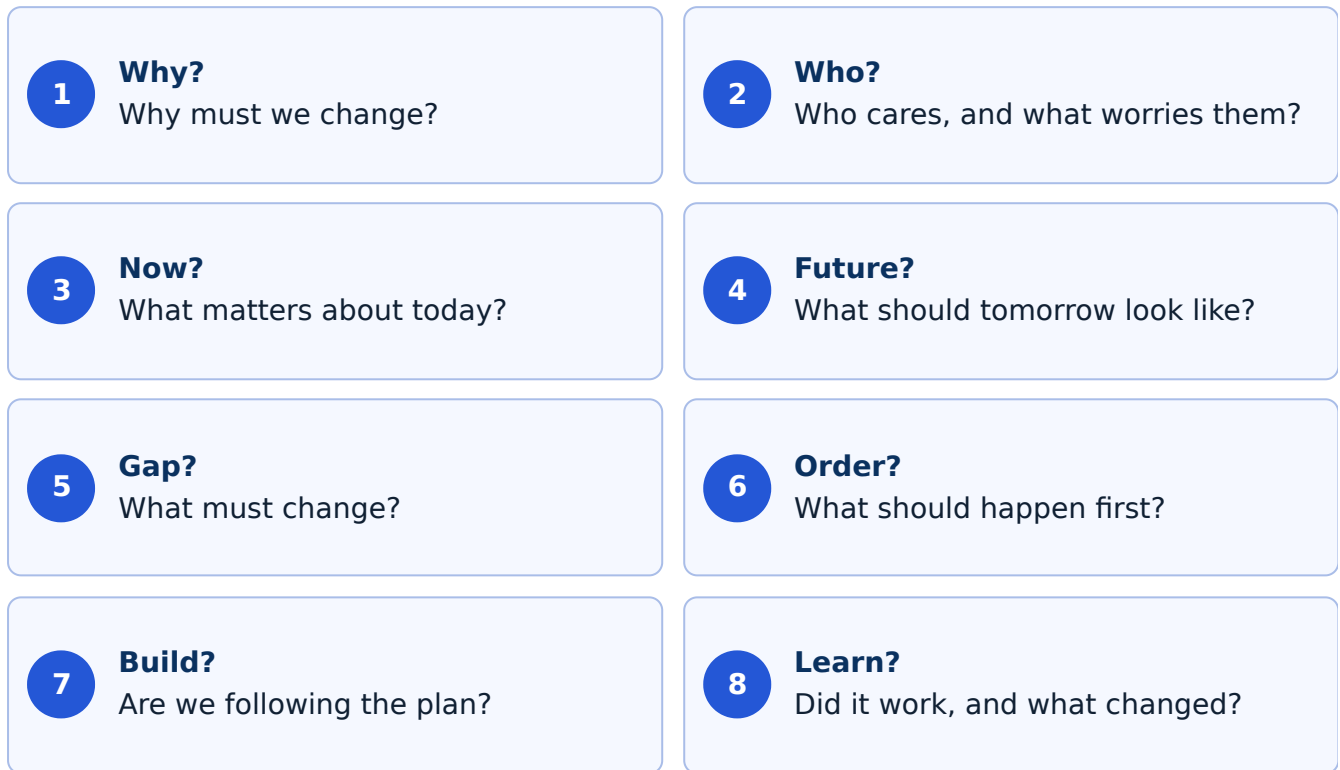
Word	Simple meaning
Enterprise	The organization or part of it that we are studying
Architecture	The important structure, relationships, and rules
Framework	An organized set of ideas and tools for doing the work

FEYNMAN CHECK

Say it back: Explain TOGAF using the town analogy. Then explain why TOGAF is not a finished town plan.

2. The whole TOGAF story

Every useful architecture journey answers the same questions:



For Sunny Toys:

1. **Why?** Customers leave because stock information is wrong.
2. **Who?** Customers, shop staff, managers, finance, security, and delivery teams all care about different things.
3. **Now?** Each shop has separate processes, data, and applications.
4. **Future?** One joined-up buying and collection experience.
5. **Gap?** Shared stock data, clearer ownership, connected applications, staff training, and reliable technology are missing.
6. **Order?** Clean stock data before promising real-time availability.
7. **Build?** Check each release against the agreed requirements.
8. **Learn?** Measure collection success, stock accuracy, customer happiness, cost, and new risks.

Memory hook: Why → Who → Now → Future → Gap → Order → Build → Learn.

FEYNMAN CHECK

Say it back: Close this page and draw the eight questions as a circle. If one question is hard to explain, that is your next study gap.

3. The ADM: TOGAF's main method

The **Architecture Development Method**, or **ADM**, is the heart of TOGAF.

It organizes architecture work into:

- a setup phase;
- eight lettered phases, A through H; and
- Requirements Management in the middle.

Use this mnemonic:

Prepare A Business; Connect Digital Execution For Governed Horizons.

That gives you **P-A-B-C-D-E-F-G-H**.

Preliminary Phase — Prepare

Simple question: How will we do architecture here?

Before planning Sunny Toys, the organization agrees:

- who makes which decisions;
- which method and documents are useful;
- which principles guide choices;
- where architecture knowledge is stored; and
- how architects work with leaders and delivery teams.

Main result: a working architecture capability—not merely an architecture team.

Phase A — Agree the Architecture Vision

Simple question: Why are we doing this, for whom, and how big is the work?

Sunny Toys agrees:

- the desired customer outcome;

- the sponsor;
- stakeholders and concerns;
- what is inside and outside scope;
- high-level baseline and target;
- success measures and risks; and
- permission to continue.

Main results: the **Architecture Vision** and **Statement of Architecture Work**.

Phase B — Business Architecture

Simple question: How should the business work?

Sunny Toys designs:

- the buy-and-collect value stream;
- shop and online capabilities;
- staff responsibilities;
- customer service;
- business rules; and
- ownership of stock accuracy.

Main result: business baseline, target, and gaps.

Phase C — Information Systems Architectures

Phase C contains two domains.

Data Architecture

Simple question: What information do we need, and who looks after it?

Sunny Toys defines:

- one meaning for Product, Shop, Stock, Customer, Order, and Collection;
- an owner for each important kind of data;
- where trusted data comes from;
- how it moves; and
- privacy, quality, retention, and access rules.

Application Architecture

Simple question: What applications and services use that information?

Sunny Toys decides how the website, stock service, payment service, shop application, notification service, and reporting tools work together.

Main result: data and application baselines, targets, and gaps.

Phase D — Technology Architecture

Simple question: What technology lets the target work safely and reliably?

Sunny Toys chooses the needed:

- platforms and hosting;
- networks and shop connectivity;
- identity and access services;
- monitoring and backup;
- integration technology; and
- technology standards and lifecycle rules.

Main result: technology baseline, target, and gaps.

Phase E — Opportunities & Solutions

Simple question: How do we turn all the gaps into sensible pieces of work?

Sunny Toys groups changes into work packages:

1. create trusted stock data;
2. connect shop systems;
3. launch online ordering;
4. add collection notifications; and
5. improve reporting and operations.

It may define **Transition Architectures**—safe intermediate states on the way to the target.

Main result: work packages and candidate transition architectures.

Phase F — Migration Planning

Simple question: What should happen first, and can we really do it?

Sunny Toys compares:

- value;
- cost;
- risk;
- dependencies;
- people and skills;
- readiness; and
- time.

Main results: a prioritized **Architecture Roadmap** and an executable **Implementation and Migration Plan**.

Phase G — Implementation Governance

Simple question: Is delivery building what was agreed?

Sunny Toys:

- agrees responsibilities and review points;
- checks important requirements;
- reviews evidence;
- handles justified exceptions; and
- avoids turning governance into a slow approval queue.

Main results: Architecture Contracts or equivalent agreements, compliance evidence, decisions, and managed deviations.

Phase H — Architecture Change Management

Simple question: Has the world changed enough to update the architecture?

Sunny Toys watches:

- business outcomes;
- customer behavior;
- operating problems;
- new laws and risks;
- technology changes; and
- architecture debt.

A small change may update one decision. A big change may start a new ADM cycle.

Main result: a living architecture instead of an old document.

Requirements Management — the center

A **requirement** says what must be true.

Example: “A customer must only see stock that can be reserved.”

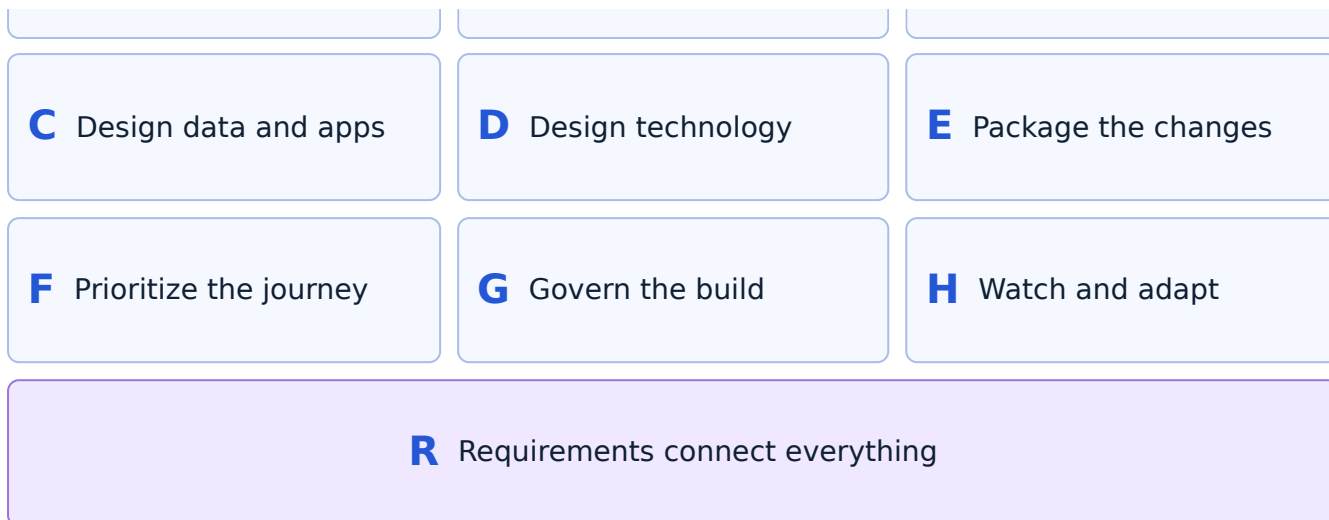
Requirements Management stays in the center because every phase:

- discovers requirements;
- uses requirements;
- tests requirements;
- changes requirements; or
- is changed by requirements.

P Prepare the practice

A Agree the vision

B Design the business



Do not treat the ADM as a waterfall. You may revisit earlier work, combine phases, run work in parallel, or use small cycles. Tailor it to the decision, risk, and delivery speed.

FEYNMAN CHECK

Say it back: Draw P through H. For each phase, say its question in one sentence. Put Requirements Management in the center and explain why.

4. The four architecture domains: BDAT

Use **BDAT**:

Domain	Child-simple question	Sunny Toys example
B — Business	How do people create value?	Buy online and collect in a shop
D — Data	What information do they need?	Product, stock, customer, order
A — Application	What software behavior helps them?	Website, stock, payment, shop app
T — Technology	What machinery lets the software run?	Cloud, network, identity, monitoring

The domains are different, but they must fit together.

If the business promises one-hour collection, then:

- Data must show trustworthy stock.
- Applications must reserve the toy.
- Technology must keep the service available.
- People and processes must prepare the collection.

Security, risk, privacy, integration, customer experience, and sustainability cross all four domains. They are not decorations added at the end.

Memory hook: Business asks for value. Data carries meaning. Applications provide behavior. Technology provides the platform.

FEYNMAN CHECK

Say it back: Pick a school, shop, or app. Give one Business, Data, Application, and Technology example. Then name one cross-cutting concern.

5. The core language

Stakeholder, concern, view, viewpoint

These four words form one chain:

1. A **stakeholder** is a person or group that cares.
2. A **concern** is what they care about.
3. A **viewpoint** is the rule for creating a useful picture.
4. A **view** is the actual picture made for this architecture.

Example:

- Stakeholder: shop manager
- Concern: “Will collection create long queues?”
- Viewpoint: show the customer and staff journey with timings
- View: Sunny Toys’ actual collection journey diagram

Baseline, target, gap, transition, roadmap

Term	Simple meaning
Baseline	The important truth about today
Target	The future we intend to create
Gap	A difference that needs action
Transition Architecture	A safe intermediate version of the enterprise
Roadmap	The ordered path through the changes

Driver, goal, outcome, requirement, measure

Term	Sunny Toys example
Driver	Customers expect online ordering
Goal	Make buying and collection easy
Outcome	More completed sales with fewer stock mistakes
Requirement	Reserved stock must not be sold to someone else
Measure	Collection success rate and stock accuracy

Principle, constraint, assumption, risk

Term	Simple meaning
Principle	A lasting rule that guides decisions
Constraint	A limit we must work within
Assumption	Something we currently believe is true
Risk	Uncertainty that may affect an outcome

Example principle: **Stock has one trusted owner.**

Example constraint: **The first release must use existing shop scanners.**

Example assumption: **Every shop has stable internet access.**

Example risk: **Old stock records may be too inaccurate for online promises.**

FEYNMAN CHECK

Say it back: Explain each chain without reading: stakeholder → concern → viewpoint → view, and baseline → target → gap → transition → roadmap.

6. What architects produce

TOGAF distinguishes three things.

Deliverable

A **deliverable** is a formally reviewed package of work.

Think: a school project handed to the teacher.

Examples:

- Architecture Vision;
- Architecture Definition Document;
- Architecture Requirements Specification;
- Architecture Roadmap; and
- Implementation and Migration Plan.

Artifact

An **artifact** is a representation of architecture information inside or alongside the work.

There are three simple forms:

Form	It answers	Example
Catalog	What exists?	Application list
Matrix	How are things related?	Capability-to-application matrix
Diagram	How should we see it?	Application interaction diagram

Building block

A **building block** is a reusable unit of capability or solution.

- An **Architecture Building Block (ABB)** says what capability is needed.
- A **Solution Building Block (SBB)** says how it will be implemented.

Sunny Toys example:

- ABB: “Secure customer identity capability”

- SBB: “The selected identity platform and its configured services”

Memory hook: A deliverable is the reviewed package. An artifact is a picture or structured list. A building block is a reusable piece of the architecture or solution.

The minimum useful architecture pack

Do not create documents because a framework has a template. For many engagements, a small useful pack is:

1. outcome, scope, sponsor, and stakeholders;
2. principles, assumptions, constraints, and risks;
3. relevant baseline;
4. target choices across the necessary domains;
5. traceable requirements and decisions;
6. gaps, work packages, and roadmap;
7. governance and exception approach; and
8. measures of success.

FEYNMAN CHECK

Say it back: Explain deliverable, artifact, ABB, and SBB using building a tree house as your example.

7. Reuse and the Architecture Repository

Imagine a kitchen.

The **Enterprise Continuum** is the labeling idea that sorts recipes and ingredients from very general to very specific.

The **Architecture Repository** is the actual pantry, cookbook shelf, fridge, and list that store and control them.

```
<strong>Enterprise  
Continuum</strong>  
<span>Classifies knowledge from  
generic to organization-specific.  
</span>
```

```
<strong>Architecture  
Repository</strong>  
<span>Stores, controls, connects,  
and helps people reuse knowledge.  
</span>
```

The Continuum has two related sides:

- **Architecture Continuum:** reusable architecture ideas and descriptions.
- **Solutions Continuum:** products, services, and implemented solutions.

Common Repository areas are:

Area	Simple contents
Architecture Metamodel	Rules about the kinds of information and relationships
Architecture Capability	Roles, method, governance, skills, and tools
Architecture Landscape	Baseline, transition, and target descriptions
Standards Information Base	Approved standards, products, versions, and lifecycle
Reference Library	Patterns, templates, reference models, and lessons
Governance Log	Decisions, reviews, exceptions, risks, and debt

Good repository content has an owner, status, version, and review date.

FEYNMAN CHECK

Say it back: Use the kitchen analogy to explain the difference between the Enterprise Continuum and Architecture Repository.

8. Governance without becoming the architecture police

Governance means important decisions are:

- made by the right people;
- based on visible criteria and evidence;
- recorded;
- followed during delivery;
- allowed a controlled exception when justified; and
- improved using feedback.

It does **not** mean one central committee designs everything.

Think of road rules:

- rules make movement safer;
- signs make expectations visible;
- drivers still choose their route;
- unusual situations have an exception process; and
- accident evidence helps improve the rules.

The governance loop

1. Agree principles and standards.
2. Define decision rights.
3. Make and record decisions.
4. Agree implementation responsibilities.
5. Review evidence of conformance.
6. Approve, correct, or manage an exception.
7. Measure the outcome.
8. Improve the architecture and rules.

An **Architecture Board** is one governance body. It needs a clear mandate, authority, decision criteria, membership, and escalation rules. Low-risk decisions should stay close to delivery.

An **Architecture Contract** is an agreement about responsibilities, requirements, evidence, reviews, and exceptions. It does not have to be a giant legal document.

FEYNMAN CHECK

Say it back: Explain why good governance is like road rules, not like one person driving every car.

9. The small toolbox of techniques

Use a technique only when it helps answer a question.

Technique	Simple purpose
Stakeholder analysis	Find who cares, why, how much influence they have, and how to involve them
Business scenario	Turn a fuzzy problem into actors, steps, outcomes, measures, and requirements
Capability-Based Planning	Plan around abilities the enterprise needs, not only projects or systems
Value-stream mapping	Follow how value moves from a need to an outcome
Gap analysis	Compare baseline and target to identify change
Readiness assessment	Check whether people, skills, culture, funding, and governance can support change
Risk management	Find uncertainty, understand exposure, choose treatment, and keep watching
Trade-off analysis	Compare options using explicit criteria and consequences
Dependency analysis	Find what relies on what before creating the sequence
Transition Architectures	Define safe intermediate states

A simple trade-off method

When Sunny Toys must choose a solution:

1. State the decision.
2. Name the stakeholders and concerns.
3. Define criteria and constraints.
4. Create more than one credible option, including “do nothing” where useful.
5. Gather enough evidence—not endless evidence.
6. Compare value, cost, risk, uncertainty, and reversibility.
7. Let the person with the decision right decide.
8. Record the choice, consequences, and revisit trigger.

FEYNMAN CHECK

Say it back: Pick one everyday choice. Compare two options using outcome, cost, risk, and reversibility.

10. TOGAF with Agile, products, DevOps, cloud, and security

TOGAF and modern delivery solve different parts of the problem.

- TOGAF helps decide the important direction and coordinate change.
- Product management keeps attention on persistent customer value.
- Agile creates short learning loops.
- DevOps makes delivery and operations fast and repeatable.
- Cloud provides on-demand technology services.
- Security and risk protect outcomes across every domain and phase.

They can work together.

Good modern architecture:

- works in small increments;
- gives teams clear guardrails and decision rights;
- records important decisions briefly;
- automates standards and evidence where possible;
- keeps architects close to products and delivery;
- uses operational data to update architecture; and
- escalates only choices with material risk or cross-enterprise impact.

Bad architecture creates a document mountain and arrives after all decisions have already been made.

Useful rule: Centralize the decisions that require enterprise coherence. Delegate the decisions that teams can safely make locally.

FEYNMAN CHECK

Say it back: Explain how TOGAF and Agile can work together. Your answer must include direction, guardrails, short feedback, and learning.

11. How to start TOGAF in a real organization

Do not announce, “We will implement the whole framework.”

Start with a real business problem.

Days 1-30: establish the minimum

- Confirm the sponsor and outcome.
- Identify stakeholders and decision rights.
- Agree a small set of principles.
- Tailor the ADM and minimum content.
- Create a simple repository structure.
- Choose one valuable pilot.

Days 31-60: run the pilot

- Agree the vision and scope.
- Develop only the useful baseline and target.
- Trace requirements and decisions.
- Identify gaps and work packages.
- Build a practical roadmap.
- Work beside delivery teams.

Days 61-90: connect and improve

- Run proportionate governance and exception handling.
- Measure value, speed, risk, and conformance.
- Reuse patterns and lessons.
- Fix unclear decision rights.
- Expand only what has proved useful.

Common failure patterns

Failure	Why it fails	Better move
Framework-first rollout	People see process, not value	Start with one outcome
Artifact factory	Documents grow; decisions do not improve	Produce only decision-useful content
Review-board bottleneck	Architecture becomes a queue	Delegate low-risk decisions
Ivory-tower target	Delivery evidence arrives too late	Work with product and delivery teams
Tool-led metamodel	The repository becomes the goal	Start with questions and users
Architecture without authority	Good advice can be ignored	Make decision rights explicit
Governance without feedback	Rules become stale	Measure outcomes and learn

FEYNMAN CHECK

Say it back: Explain how you would begin TOGAF without saying, “First, create every TOGAF document.”

12. One full Sunny Toys walkthrough

Here is the entire method in one short story.

Prepare

Sunny Toys gives the transformation sponsor authority. A small Architecture Board handles only cross-shop decisions. Product teams keep local delivery choices within shared guardrails.

Agree

The vision is: “Customers can confidently buy online and collect from any shop within two hours.” Scope includes stock, ordering, payment, collection, staff workflow, and enabling technology. It excludes home delivery for now.

Understand the business

The target value stream is:

Find toy → reserve → pay → prepare → notify → collect → support

Shop staff own physical stock accuracy. Customer service owns collection support. Product leaders own the end-to-end outcome.

Design information and applications

Product, Shop, Stock, Customer, Order, Payment, and Collection get clear definitions and owners. A stock service becomes the trusted source. The website and shop application use the same order and stock services.

Design technology

The target uses secure identity, managed hosting, reliable shop connectivity, monitoring, backup, and standard integration. Privacy and security requirements apply throughout.

Package and plan

Sunny Toys creates three transition states:

1. accurate shared stock;
2. online reservation with payment at collection; and
3. full online payment and two-hour collection.

The roadmap follows data dependency and business readiness, not excitement about the website.

Govern and learn

Each release supplies automated test, security, reliability, and stock-accuracy evidence. Exceptions have an owner and expiry. Sunny Toys measures customer completion, stock accuracy, collection time, support calls, cost, and incidents.

Those results update the architecture.

FEYNMAN CHECK

Say it back: Retell the Sunny Toys story without phase letters. Then add the phase letters to each step.

13. The exam idea, simply

The current Enterprise Architecture learning path has two main levels:

Foundation

Foundation checks whether you understand:

- core concepts and vocabulary;
- the ADM structure and purpose of each phase;
- foundational techniques;
- applying the ADM;
- governance; and
- architecture content.

The Part 1 examination uses 40 multiple-choice questions in 60 minutes, with a 60% passing score.

Practitioner

Practitioner checks whether you can apply the framework to a situation.

The Part 2 examination uses eight scenario-based questions in 90 minutes. Each question has four choices scored 5, 3, 1, or 0. It is open book using the provided reference material.

Use SCOPE for scenarios

1. **S — Stakeholders:** Who cares and who decides?
2. **C — Context:** What phase, scope, evidence, and constraints exist?
3. **O — Outcome:** What result or decision is needed?
4. **P — Proportion:** What is the smallest sufficient response?
5. **E — Evidence:** Which choice preserves traceability, trade-offs, and feedback?

Prefer answers that clarify outcomes and authority, address concerns, tailor the work, maintain traceability, and enable governed delivery.

Be suspicious of answers that:

- create every artifact;
- skip straight to technology;
- centralize every choice;
- ignore stakeholder concerns; or
- treat the ADM as a fixed waterfall.

FEYNMAN CHECK

Say it back: Explain the difference between Foundation and Practitioner. Then use SCOPE on a decision from the Sunny Toys story.

14. One-page memory sheet

TOGAF in one sentence

TOGAF helps an organization understand where it is, agree where it wants to go, plan the changes, govern delivery, and keep learning.

The journey

Why → Who → Now → Future → Gap → Order → Build → Learn

The ADM

Phase	Remember
Preliminary	Prepare the architecture practice
A	Agree vision, value, scope, and authority
B	Design how the business creates value
C	Design data and applications
D	Design enabling technology
E	Package gaps into work
F	Prioritize the migration journey
G	Govern implementation
H	Watch change and adapt
Requirements	Connect and trace everything

The domains

BDAT = Business, Data, Application, Technology

The content

- Deliverable = reviewed package
- Artifact = catalog, matrix, or diagram
- ABB = required capability
- SBB = implementation

Reuse

- Continuum = classifies
- Repository = stores and controls

Governance

Decision rights → decision → agreement → evidence → compliance or exception → outcome → feedback

Scenario thinking

SCOPE = Stakeholders, Context, Outcome, Proportion, Evidence

FEYNMAN CHECK

Final say-it-back: Close the book. Recreate this memory sheet on one blank page. Teach it to another person in ten minutes.

15. Tiny glossary

Term	Plain meaning
ABB	What architecture capability is required
ADM	TOGAF's main iterative method
Architecture Board	Body accountable for defined architecture decisions
Architecture Contract	Agreement about implementation and conformance
Architecture Repository	Managed home for architecture knowledge
Artifact	Catalog, matrix, diagram, or other representation
Baseline	Relevant current state
Building block	Reusable architecture or solution component
Business capability	What the enterprise is able to do
Concern	What matters to a stakeholder
Deliverable	Formally reviewed work product
Enterprise	The organization or scope being studied
Enterprise Continuum	Way to classify assets from generic to specific
Gap	Difference between baseline and target that needs action
Governance	How decisions, controls, exceptions, and feedback work
Principle	Durable rule for making decisions
Requirement	Condition that must be satisfied
SBB	Specific implementation of required capability
Stakeholder	Person or group with an interest
Target	Intended future state
TOGAF	Framework for developing and governing Enterprise Architecture

Term	Plain meaning
Transition Architecture	Coherent intermediate state
Value stream	Stages that create value for a stakeholder
View	Actual representation for stakeholder concerns
Viewpoint	Rules for constructing and using a view
Work package	Bounded set of change actions

16. Prove that you understand

Answer these aloud in simple words:

1. What problem does Enterprise Architecture solve?
2. What does TOGAF give you?
3. Why is the ADM not a waterfall?
4. What question does each ADM phase answer?
5. Why is Requirements Management in the center?
6. How do Business, Data, Application, and Technology fit together?
7. What is the difference between a view and a viewpoint?
8. What is the difference between baseline, target, gap, transition, and roadmap?
9. What is the difference between a deliverable, artifact, ABB, and SBB?
10. Why is the Enterprise Continuum not the Architecture Repository?
11. What makes governance useful instead of slow?
12. How can TOGAF work with Agile and DevOps?
13. How would you start TOGAF in an organization?
14. How does SCOPE help with a scenario?
15. Can you retell the Sunny Toys story from problem to learning?

If an answer needs lots of jargon, simplify it. If you cannot simplify it, find the small gap and study it again. That is the Feynman Technique.

You understand TOGAF when you can:

- draw the whole journey from memory;
- explain each phase with a simple question;
- apply it to a real change;
- choose only the useful content and techniques; and
- show how decisions lead to outcomes and learning.

Continue learning

- Read the full TOGAF Handbook for detail.
- Use the fast-learning mind map for recall.
- Open the editable draw.io workbook to redraw the relationships.
- Take the timed mock exam and review mistakes by concept.
- Use the comprehensive glossary whenever a word becomes fuzzy.

Official sources

This is an independent educational summary. It does not reproduce or replace the official standard.

- [TOGAF overview](#)
- [TOGAF Standard publications catalog](#)
- [TOGAF Library](#)
- [TOGAF certification portfolio](#)

TOGAF Learning and Syllabus Coverage Matrix

This matrix traces the published TOGAF Enterprise Architecture Foundation and Practitioner topic areas to the learning and practice assets in this repository. It is aligned to the TOGAF Standard, 10th Edition learning path.

The authoritative learning outcomes are defined in [The Open Group document X2202](#). This matrix paraphrases topic coverage; it does not reproduce protected syllabus or examination content.

Coverage status

Status	Meaning
Complete	Explanation, recall practice, and assessment evidence are present
Applied	A scenario or worked example requires application of the topic
Reference	Supporting detail is available for lookup

Foundation coverage

The official OGEA-101 plan contains 40 questions across six areas: [8 Concepts](#), [14 ADM](#), [6 ADM Techniques](#), [4 Applying the ADM](#), [3 Governance](#), and [5 Architecture Content](#).

ID	Published area	Exam weight	Knowledge and recall coverage	Practice evidence	Status
F1	Concepts	8/40	Executive overview , Core concepts , Continuum and Repository , Glossary	Flashcards, all four Foundation mock sets, readiness assessment	Complete
F2	Introduction to the ADM	14/40	ADM deep dive , Simple guide , Mind map	ADM flashcards, all four Foundation mock sets, redraw check	Complete
F3	Introduction to ADM Techniques	6/40	Architecture techniques	Technique flashcards, case workbook, all four mock sets	Complete
F4	Introduction to Applying the ADM	4/40	Practical adoption , Modern delivery	Tailoring and iteration questions, cases 1 and 8	Complete
F5	Introduction to Architecture Governance	3/40	Capability and governance	Governance flashcards, mock questions, cases 1, 5, and 6	Complete
F6	Architecture Content	5/40	Content and artifacts , Templates	Content flashcards, artifact questions, case outputs	Complete

Foundation competency checklist

Use this checklist for retrieval practice. You cover an item only when you can explain it without reading.

Competency	Explain	Apply	Assessment
Purpose and value of Enterprise Architecture	✓	✓	Mock sets A-D
What TOGAF provides and does not prescribe	✓	✓	Mock sets A-D
Modular 10th Edition structure	✓	—	Flashcards and mocks
Enterprise, architecture, framework, and architecture description	✓	✓	Flashcards and mocks
Stakeholder, concern, view, and viewpoint	✓	✓	Case 2 and mocks
Baseline, target, gap, transition, and roadmap	✓	✓	Cases 3-6 and mocks
Deliverable, artifact, catalog, matrix, and diagram	✓	✓	Cases 3-5 and mocks
ABB, SBB, requirement, principle, and building block	✓	✓	Flashcards and mocks
Preliminary Phase and Architecture Capability	✓	✓	Case 1 and mocks
Phase A: vision, scope, value, stakeholders,	✓	✓	Case 3 and mocks

Competency	Explain	Apply	Assessment
and authorization			
Phase B: Business Architecture	✓	✓	Case 4 and mocks
Phase C: Data and Application Architectures	✓	✓	Case 4 and mocks
Phase D: Technology Architecture	✓	✓	Case 4 and mocks
Phase E: work packages and Transition Architectures	✓	✓	Case 5 and mocks
Phase F: priorities, dependencies, and migration planning	✓	✓	Case 5 and mocks
Phase G: contracts, compliance, and exceptions	✓	✓	Case 5 and mocks
Phase H: change assessment and new ADM cycles	✓	✓	Case 6 and mocks
Requirements Management across every phase	✓	✓	Case 7 and mocks
Stakeholder, business-scenario, capability, gap, risk, and migration techniques	✓	✓	Cases 2-8 and mocks

Competency	Explain	Apply	Assessment
Iteration, partitioning, levels, tailoring, and scope	✓	✓	Cases 1 and 8
Architecture governance, decision rights, and Architecture Board	✓	✓	Cases 1, 5, and 6
Enterprise Continuum and Architecture Repository	✓	✓	Flashcards and mocks
Content metamodel and repository areas	✓	✓	Flashcards and mocks

Practitioner coverage

The official OGEA-102 plan uses eight scenario-based questions with gradient scoring. Questions are drawn from eight major topic areas: [the EA context](#), [stakeholders](#), [Phase A](#), [Phases B-D](#), [Phases E-G](#), [Phase H](#), [Requirements Management](#), and [supporting ADM work](#).

ID	Published area	Main learning evidence	Applied evidence	Assessment evidence	Status
P1	The Context for Enterprise Architecture	Core concepts , Practical adoption	Workbook case 1	Practitioner sets A-D and readiness scenarios	Applied
P2	Stakeholder Management	Core concepts , Techniques	Workbook case 2, stakeholder-map template	Practitioner sets A-D and readiness scenarios	Applied
P3	Phase A: The Starting Point	ADM Phase A , Architecture Vision template	Workbook case 3	Practitioner sets A-D and readiness scenarios	Applied
P4	Architecture Development: Phases B, C, D	ADM , Domains	Workbook case 4, worked example	Practitioner sets A-D and readiness scenarios	Applied
P5	Implementing the Architecture: Phases E, F, G	ADM , Governance	Workbook case 5, roadmap and compliance templates	Practitioner sets A-D and readiness scenarios	Applied
P6	Architecture Change Management: Phase H	ADM Phase H , Modern delivery	Workbook case 6	Practitioner sets A-D and readiness scenarios	Applied
P7	Requirements Management	ADM Requirements , requirement template	Workbook case 7	Practitioner sets A-D and readiness scenarios	Applied
P8	Supporting the ADM Work	Techniques , Practical adoption , Modern delivery	Workbook case 8	Practitioner sets A-D and readiness scenarios	Applied

Practitioner decision checklist

For every scenario, verify that you can:

- identify the architecture context, scope, level, and decision required;
- identify stakeholders, concerns, influence, and decision rights;

- select the ADM phase that owns the immediate problem;
- use baseline evidence without documenting irrelevant detail;
- develop coherent Business, Data, Application, and Technology choices;
- turn gaps into work packages and Transition Architectures;
- prioritize using value, cost, risk, dependency, readiness, and resources;
- govern conformance while allowing controlled exceptions;
- distinguish maintenance, a partial ADM cycle, and a new cycle;
- trace requirements from source through decision, implementation, and proof;
- tailor techniques and content to risk and decision value; and
- integrate architecture with product, Agile, DevOps, security, portfolio, and operations.

Asset-to-outcome traceability

Asset	Knowledge	Retrieval	Application	Timed evidence
Full handbook	✓	—	✓	—
Simple Feynman PDF	✓	✓	✓	—
Markdown and draw.io mind maps	✓	✓	—	—
Comprehensive glossary	✓	✓	—	—
Spaced-repetition flashcards	✓	✓	—	✓
Worked example	✓	✓	✓	—
Practitioner case-study workbook	✓	✓	✓	—
Four Foundation mock sets	✓	✓	—	✓
Four Practitioner mock sets	✓	✓	✓	✓
Final readiness assessment	✓	✓	✓	✓

Completion rule

Coverage on paper is not mastery. Mark an area ready only when:

1. you can explain it in plain language without notes;
2. you can apply it to an unfamiliar scenario;
3. you can distinguish it from commonly confused terms; and
4. recent timed evidence meets the target score.

The learning dashboard combines these signals into the final readiness result.

Authoritative references

- [Official X2202 syllabus location](#)
- [OGEA-101 exam plan](#)
- [OGEA-102 exam plan](#)
- [Foundation versus Practitioner competencies](#)

TOGAF Practitioner Case-Study Workbook

These eight original cases mirror the eight published Practitioner topic areas. They do not reproduce protected examination questions.

How to work each case

Use **SCOPE** before writing an answer:

1. **Stakeholders** — who cares and who decides?
2. **Context** — what phase, level, scope, evidence, and constraints exist?
3. **Outcome** — what result or decision must the work enable?
4. **Proportion** — what is the smallest sufficient response?
5. **Evidence** — what preserves traceability, trade-offs, and feedback?

For each case:

- write your answer before reading the model response;
 - explain why at least one tempting response is weaker;
 - name the ADM phase or supporting technique involved; and
 - state what evidence would prove success.
-

Case 1: Establishing the Enterprise Architecture Context

Scenario

Northstar Foods has acquired four regional manufacturers. Each business has its own product catalog, supply-chain processes, applications, infrastructure, and technology standards. The board wants lower operating cost and a single view of inventory. The new CIO creates a five-person Enterprise Architecture team and asks it to “standardize everything within six months.”

Business presidents fear losing their ability to serve local markets. Product teams worry that a new central review board will delay releases. Finance wants fast savings, while operations warns that several factories rely on old equipment that cannot be replaced immediately. No

architecture mandate, decision-right model, repository, or connection to portfolio funding exists.

Your tasks

1. What is the real problem behind the request to standardize everything?
2. What should the lead architect establish before launching broad domain work?
3. How should decision rights balance enterprise coherence with local autonomy?
4. What would make a good first pilot?
5. Name three measures that would show the architecture capability is useful.

Worksheet

- Outcome:
 - Scope and level:
 - Key stakeholders and concerns:
 - First architecture decision:
 - Minimum content:
 - Governance and feedback:
-

Case 2: Stakeholder Management

Scenario

CityLink, a public transport authority, is designing one mobile ticket across bus, rail, and ferry services. The customer director wants an effortless journey. The finance director wants fraud controls. Accessibility groups need the service to work for passengers with disabilities. The privacy officer worries about travel histories. Operations teams need offline validation when connectivity fails. Private transport partners want to keep control of their customer relationships.

The sponsor asks the architecture team for a target design in three weeks. Engineering proposes creating an application diagram first because it is the fastest artifact to produce.

Your tasks

1. Why is an application diagram a weak starting point?
2. Create a stakeholder-and-concern map for the case.
3. Which viewpoints would help resolve the most important conflicts?
4. Who should decide the trade-off between convenience, privacy, and fraud?
5. How should agreed concerns become traceable requirements?

Worksheet

- Highest-influence stakeholders:
- Highest-impact stakeholders:
- Conflicting concerns:

- Required views:
 - Decision owner:
 - Traceability path:
-

Case 3: Phase A — The Starting Point

Scenario

ClearWater Utilities wants predictive maintenance for pumps across 80 sites. Executives agree that unexpected failures are costly, but they disagree about the engagement boundary. Operations wants new field processes and sensors. Data leaders want ownership and quality rules. Technology leaders want a cloud analytics platform. Procurement wants to start vendor selection immediately.

The program manager asks the architect to begin Technology Architecture so a platform choice can “make the scope real.” There is no agreed success measure, stakeholder map, readiness view, or Statement of Architecture Work.

Your tasks

1. What should happen before detailed Technology Architecture?
2. Define the business outcome and three measurable success indicators.
3. Describe the scope using breadth, depth, time horizon, and domains.
4. What belongs in the Architecture Vision?
5. What must the Statement of Architecture Work authorize and control?

Worksheet

- Driver and desired outcome:
 - Scope:
 - Stakeholders:
 - High-level baseline and target:
 - Risks and readiness:
 - Authorization required:
-

Case 4: Developing Business, Data, Application, and Technology Architectures

Scenario

Harbor Health operates twelve clinics. A patient may register separately at each clinic, repeat medical history, and receive conflicting appointment messages. Leaders want a joined-up patient journey without replacing every system at once.

Clinic managers want local flexibility. Clinicians need safe and timely information. Patients want control over communications. The data team has found four patient identifiers and no clear owner for contact data. Applications exchange nightly files. Several clinics have unreliable network links.

Your tasks

1. Describe a target Business Architecture for the patient journey.
2. Identify the most important Data Architecture choices.
3. Define the Application Architecture responsibilities without choosing products prematurely.
4. Identify Technology Architecture enablers and constraints.
5. Show two cross-domain dependencies that must remain visible.

Worksheet

- Target value stream:
 - Capabilities and ownership:
 - Data entities and owners:
 - Application services:
 - Technology services:
 - Cross-cutting security, privacy, and resilience:
-

Case 5: Implementing the Architecture — Phases E, F, and G

Scenario

Beacon Retail has approved a target architecture for unified customer orders and stock. The gap list contains 70 changes across data, applications, infrastructure, operations, training, and supplier contracts. Every business unit calls its own changes “critical.” The transformation budget can fund only half the work this year.

One team proposes building the new customer application immediately. Another warns that stock data is too unreliable to support the promised experience. Delivery teams ask what evidence will be needed at architecture reviews.

Your tasks

1. How should Phase E turn the gap list into coherent work packages?
2. Propose useful Transition Architectures.
3. What evidence should Phase F use to prioritize the roadmap?
4. How should Phase G connect architecture requirements to delivery?
5. What information must a justified exception contain?

Worksheet

- Work packages:
 - Dependencies:
 - Transition states:
 - Prioritization criteria:
 - Compliance evidence:
 - Exception rule:
-

Case 6: Phase H — Architecture Change Management

Scenario

Atlas Insurance implemented a cloud claims architecture eighteen months ago. Operational evidence now shows high cost during catastrophe events. A new privacy law limits data retention. A machine-learning service can improve fraud detection, but introduces explainability and model-risk concerns. One delivery team also requests a minor API version change.

Leaders disagree about whether every change should launch a full ADM cycle. The repository contains the approved target, but decisions and exceptions have not been updated since launch.

Your tasks

1. Classify the four change signals by likely response.
2. Which change may be ordinary maintenance?
3. Which changes justify a partial or new ADM cycle?
4. What Architecture Landscape and Governance Log updates are needed?
5. Which outcome and risk measures should be monitored?

Worksheet

- Change signal:
 - Materiality:
 - Scope of response:
 - Decision authority:
 - Repository update:
 - Revisit trigger:
-

Case 7: Requirements Management

Scenario

GreenGrid is modernizing the systems that balance renewable energy supply and demand. Requirements are kept in separate spreadsheets by engineering, security, operations, and suppliers. Many statements have no owner, source, priority, or verification method. A new regulator rule arrives after the target architecture has been approved.

The program manager wants to add the rule to one spreadsheet and continue delivery. The chief architect suspects it may affect data retention, operational processes, supplier contracts, platform capacity, and the migration sequence.

Your tasks

1. What attributes should every controlled requirement have?
2. How should the new rule enter Requirements Management?
3. What should a requirements impact assessment examine?
4. How should traceability reach implementation evidence?
5. When should architecture work be revisited?

Worksheet

- Requirement statement and source:
 - Owner and priority:
 - Affected stakeholders and domains:
 - Affected decisions and work packages:
 - Verification method:
 - Change history and approval:
-

Case 8: Supporting the ADM in an Agile Digital Enterprise

Scenario

BrightLearn runs a digital education platform with 30 autonomous product teams. Teams deploy many times per day. Enterprise architects currently review large solution documents at the end of delivery, so teams see architecture as slow and irrelevant. Shared identity, learner data, AI safety, and platform cost need enterprise coherence, while most user-interface choices do not.

The CTO wants architecture integrated into product planning and delivery without creating a central approval queue.

Your tasks

1. How should the ADM be tailored for short product learning cycles?
2. Which decisions should be centralized, federated, or delegated?

3. Which lightweight artifacts and automated evidence are useful?
4. How should security, AI risk, and cost be integrated across domains?
5. How can operational evidence feed Phase H and future roadmaps?

Worksheet

- Architecture level and cadence:
 - Enterprise guardrails:
 - Delegated decisions:
 - Minimum artifacts:
 - Automated evidence:
 - Feedback measures:
-

Model responses

Use these to find gaps in your reasoning, not to memorize sentences.

Model response 1: Context

The real problem is not a lack of standards; it is missing enterprise direction, authority, portfolio connection, and a safe way to balance shared outcomes with regional needs. Start by establishing a minimum architecture capability: mandate, sponsor, scope, tailored method, principles, roles, decision rights, repository, and integration with portfolio and delivery governance.

Use a federated model. Enterprise authority owns decisions with material cross-business impact—shared inventory meaning, integration, security, and technology lifecycle. Regions retain bounded market and delivery choices.

A strong pilot has visible business value, manageable scope, willing stakeholders, reusable learning, and real dependencies. One inventory flow across two businesses is better than an enterprise-wide documentation exercise. Measure decision time, reuse, delivery rework, risk reduction, outcome progress, and stakeholder trust.

Model response 2: Stakeholders

Start with stakeholder analysis because the architecture exists to address concerns and enable decisions. Map influence, impact, decision authority, and engagement. Useful viewpoints include customer journey and accessibility, privacy and information lifecycle, fraud and control, offline operational resilience, partner responsibility, and financial value.

The sponsor should ensure the relevant accountable authorities resolve the trade-off with evidence; engineering alone should not decide it. Turn resolved concerns into owned, measurable requirements and trace them to target choices, work packages, tests, operational measures, and review.

Model response 3: Phase A

Use Phase A before detailed domain development. Agree the outcome—fewer unplanned failures and safer, more predictable maintenance—then define measures such as outage hours, maintenance lead time, and failure-detection accuracy.

Scope the sites and business processes, required detail, planning horizon, and Business, Data, Application, and Technology domains. The Vision should show the problem, stakeholders, high-level baseline and target, value, requirements, constraints, risks, and readiness. The Statement of Architecture Work authorizes objectives, scope, method, plan, roles, governance, deliverables, acceptance, assumptions, and change control.

Model response 4: Architecture Development

The Business target is one joined-up patient journey with clear clinical, administrative, and communication responsibilities while preserving justified local variation. Data choices include patient identity resolution, ownership, consent, quality, lifecycle, and a trusted contact source.

Application Architecture should define services such as patient identity, appointment, notification, and clinical-information access before products. Technology must enable secure identity, resilient connectivity, integration, monitoring, and recovery. Cross-domain dependencies include consent rules driving data and notification behavior, and connectivity constraints affecting application resilience and clinic process.

Model response 5: Implementation

Phase E groups related gaps into outcome-oriented work packages, reconciles cross-domain dependencies, considers solution building blocks, and defines safe Transition Architectures. A sensible sequence establishes trustworthy stock before customer promises, then reservation, then richer fulfillment.

Phase F uses value, cost, risk, dependency, readiness, resources, and delivery evidence—not stakeholder volume—to prioritize. Phase G agrees responsibilities, review points, and required automated or manual evidence. An exception records exact scope, rationale, risk, compensating controls, owner, authority, expiry, and remediation or exit.

Model response 6: Change

The API version may be maintenance if bounded and compatible. The cost problem may need targeted architecture work. The privacy law has mandatory cross-domain impact and may require a partial cycle. Machine learning may require a new or expanded cycle because it changes capability, data, technology, risk, and governance.

Update relevant baseline and target states, requirements, decisions, standards, exceptions, risks, and debt. Monitor cost, performance, compliance, model quality, explainability, incidents, and business outcomes. Use materiality, scope, urgency, and risk to choose maintenance, partial iteration, or a new cycle.

Model response 7: Requirements

Each requirement needs an identifier, clear statement, rationale, source, owner, priority, status, acceptance or verification method, relationships, and change history. Register the regulation as a sourced requirement, analyze it, and assess its effect on stakeholders, domains, architecture decisions, gaps, work packages, risk, cost, and sequence.

Trace it from source to requirement, decision, architecture element, work package, implementation control, test or review evidence, and operational measure. Revisit affected ADM work whenever the impact invalidates approved assumptions, target choices, roadmap, or governance.

Model response 8: Supporting the ADM

Use small ADM cycles aligned with product horizons and continuous discovery. Centralize cross-enterprise identity, learner-data semantics, AI safety, and material platform-risk decisions. Federate domain choices and delegate local, reversible user-interface decisions.

Use brief visions, principles, decision records, domain contracts, roadmaps, and machine-checkable standards. Automate evidence through delivery pipelines, security tests, policy checks, observability, and cost controls. Feed incidents, performance, usage, risk, cost, and customer outcomes into Phase H and future planning.

Scoring rubric

Score each case from 0 to 4:

	Score	Evidence
	0	Repeats words without identifying the decision
	1	Identifies a relevant phase or technique
	2	Connects stakeholders, outcome, and architecture work
	3	Handles trade-offs, traceability, governance, and evidence
	4	Gives a proportionate response and explains why alternatives are weaker

A strong readiness target is at least **3 on every case**, not merely a high average.

TOGAF Abbreviations and Terminology Glossary

This is a comprehensive learning glossary for the TOGAF Standard, 10th Edition and the Enterprise Architecture terminology commonly encountered around it. Definitions are concise, original explanations rather than reproduced normative text.

Use the labels as follows:

- **Core TOGAF** — central vocabulary used by the current framework;
- **Common EA** — widely used Enterprise Architecture or delivery language, but not necessarily a prescribed TOGAF term;
- **Certification** — examination or learning-path terminology; and
- **Classic TOGAF** — terminology frequently found in earlier TOGAF material or reference models and still encountered in practice.

For exact normative wording, consult the [official TOGAF publications catalog](#).

Abbreviations and acronyms

Abbreviation	Expanded form	Meaning and usage	Status
ABB	Architecture Building Block	A reusable description of a required architecture capability or behavior, independent of a particular implementation.	Core TOGAF
ACMM	Architecture Capability Maturity Model	A model for assessing and improving the maturity of an architecture capability.	Common EA
ACR	Architecture Change Request	A request to assess or change an approved architecture, requirement, roadmap, or governed implementation.	Core TOGAF
ADD	Architecture Definition Document	Common shorthand for the integrated description of baseline, target, gaps, rationale, and relevant domain views.	Common shorthand
ADM	Architecture Development Method	TOGAF's central iterative method for developing, governing, and evolving architecture.	Core TOGAF
ADR	Architecture Decision Record	A short record of a consequential decision, its context, options, outcome, and consequences.	Common EA
API	Application Programming Interface	A defined contract through which software components interact. APIs are often application or integration architecture subjects.	Common technology
ARB	Architecture Review Board	A common industry name for a body	Common EA

Abbreviation	Expanded form	Meaning and usage	Status
		reviewing architecture decisions. TOGAF more commonly uses Architecture Board .	
ARS	Architecture Requirements Specification	Common shorthand for the controlled set of measurable requirements an architecture and its implementations must satisfy.	Common shorthand
BA	Business Architecture	The architecture domain concerned with strategy, capabilities, value, organization, services, and processes. Avoid confusing it with Business Analyst .	Common shorthand
BDAT	Business, Data, Application, Technology	A memory aid for the four commonly used architecture domains.	Common EA
BoK	Body of Knowledge	The material forming the knowledge base for a profession, standard, or certification.	Common learning term
BPMN	Business Process Model and Notation	A standard notation for modeling business processes; it may be used with TOGAF but is not required by it.	Related standard
BTRA	Business Transformation Readiness Assessment	Assessment of factors that enable or obstruct successful transformation.	Common shorthand
CBP	Capability-Based Planning	Planning change around the business capabilities needed to achieve strategic outcomes.	Common shorthand
CI/CD	Continuous Integration and Continuous Delivery or Deployment	Automated delivery practices that architecture governance should	Common delivery

Abbreviation	Expanded form	Meaning and usage	Status
		enable and integrate with.	
COTS	Commercial Off-the-Shelf	A product acquired rather than built specifically for the organization.	Common technology
EA	Enterprise Architecture	The discipline connecting strategic intent and outcomes to coordinated enterprise change.	Core concept
ERP	Enterprise Resource Planning	Integrated application platforms for functions such as finance, procurement, supply chain, or human resources.	Common technology
IAM	Identity and Access Management	Capabilities controlling identities, authentication, authorization, and access lifecycle.	Common technology
IFADM	Implementation Factor Assessment and Deduction Matrix	A structured migration-planning technique for recording implementation factors, impacts, deductions, and resulting actions.	Common shorthand
III-RM	Integrated Information Infrastructure Reference Model	A classic TOGAF reference model concerned with integrated information infrastructure and Boundaryless Information Flow.	Classic TOGAF
IMP	Implementation and Migration Plan	Common shorthand for the prioritized, resourced, executable plan for delivering work packages and transition states.	Common shorthand
IS	Information Systems	In ADM Phase C, the combination of Data	Core TOGAF

Abbreviation	Expanded form	Meaning and usage	Status
		Architecture and Application Architecture.	
IT	Information Technology	Technology used to create, process, store, exchange, secure, and operate information and services.	General
ITSM	Information Technology Service Management	Practices for designing, delivering, operating, and improving technology services.	Related practice
KPI	Key Performance Indicator	A measure used to evaluate performance against an intended result.	Common management
NFR	Non-Functional Requirement	A quality or constraint such as security, reliability, performance, accessibility, or maintainability.	Common engineering
OGEA-101	TOGAF Enterprise Architecture Part 1 examination	The examination associated with the Foundation learning path.	Certification
OGEA-102	TOGAF Enterprise Architecture Part 2 examination	The scenario-based examination associated with the Practitioner learning path.	Certification
OGEA-103	TOGAF Enterprise Architecture Combined Part 1 and Part 2 examination	The combined examination containing the Part 1 and Part 2 sections.	Certification
OKR	Objectives and Key Results	A goal-setting method that can provide strategic outcomes and measures for architecture work.	Common management
PMO	Project or Portfolio Management Office	A function coordinating projects or investment portfolios; architecture should connect to its	Common management

Abbreviation	Expanded form	Meaning and usage	Status
		planning and governance processes.	
RACI	Responsible, Accountable, Consulted, Informed	A responsibility-assignment model useful for clarifying architecture roles and decision participation.	Common management
RFAW	Request for Architecture Work	Common shorthand for the formal trigger or request to begin architecture work.	Common shorthand
ROI	Return on Investment	A comparison of expected or realized benefit with investment cost.	Common management
SBB	Solution Building Block	A specific implementation component that realizes one or more architecture building blocks.	Core TOGAF
SIB	Standards Information Base	The repository area holding adopted standards, products, versions, lifecycle status, and related guidance.	Core TOGAF
SLA	Service-Level Agreement	An agreement defining measurable service expectations such as availability, response, or support.	Common service management
SME	Subject-Matter Expert	A person with deep knowledge of a business, data, application, technology, regulatory, or delivery subject.	Common
SOA	Service-Oriented Architecture	An architecture style organizing capabilities as interoperable services.	Common architecture

Abbreviation	Expanded form	Meaning and usage	Status
SoAW	Statement of Architecture Work	Informal shorthand for the agreement defining an architecture engagement's scope, approach, plan, governance, and acceptance.	Common shorthand
SWOT	Strengths, Weaknesses, Opportunities, Threats	A strategic analysis technique that may help explore context and drivers.	Common management
TOGAF	The Open Group Architecture Framework	The name originated from this phrase. The current standard is normally referred to simply as the TOGAF Standard .	Core TOGAF
TRM	Technical Reference Model	A classic TOGAF reference model and taxonomy of generic platform services.	Classic TOGAF
UML	Unified Modeling Language	A general-purpose software and systems modeling language; TOGAF does not mandate it.	Related standard

ADM phase terminology

Phase or activity	Purpose	Typical results
Preliminary Phase	Establish how the organization will perform and govern Enterprise Architecture.	Tailored method, principles, roles, decision rights, repository, and architecture capability arrangements
Phase A: Architecture Vision	Agree the problem, outcomes, stakeholders, value, scope, and authorization for the engagement.	Architecture Vision and approved Statement of Architecture Work
Phase B: Business Architecture	Describe how the business should operate to achieve the agreed outcomes.	Business baseline, target, gaps, capabilities, value streams, organization, services, and processes
Phase C: Information Systems Architectures	Develop the Data and Application Architectures needed by the business target.	Data and application baselines, targets, gaps, services, entities, ownership, and interactions
Phase D: Technology Architecture	Develop the technology environment that enables the target information systems.	Technology baseline, target, gaps, platforms, infrastructure services, and standards
Phase E: Opportunities & Solutions	Combine gaps into coherent work packages and identify useful transition states.	Work packages, candidate solutions, dependencies, and transition architectures
Phase F: Migration Planning	Prioritize work and create a viable, integrated delivery plan.	Architecture Roadmap and Implementation and Migration Plan
Phase G: Implementation Governance	Keep delivery aligned with the approved architecture while managing justified deviations.	Architecture Contracts, compliance assessments, decisions, and exceptions
Phase H: Architecture Change Management	Monitor change and determine whether the architecture or ADM cycle must evolve.	Change assessments, Architecture Change Requests, updates, or a new ADM cycle
Requirements Management	Continuously elicit, analyze, prioritize, trace, verify, and control requirements.	Requirements repository, traceability, and impact feedback across every ADM phase

A-Z terminology

A

Acceptance criteria — Conditions used to decide whether a deliverable, architecture, implementation, or transition has been completed acceptably. Clear criteria are observable and linked to outcomes or requirements.

Actor — A person, organization, role, or system that performs behavior or interacts with a service or process.

Agile — A family of iterative, feedback-driven approaches to developing products and services. TOGAF can work with Agile delivery when architecture is incremental, proportional, and connected to product decisions.

Application Architecture — The domain covering application services, responsibilities, interactions, interfaces, portfolio structure, and alignment to business and data needs.

Architecture — The fundamental organization of a subject and the principles guiding its evolution; in practice, also the discipline of making and governing consequential structural choices.

Architecture Board — An accountable governance body with authority over defined architecture decisions, standards, conformance, and exceptions. It need not review every design.

Architecture Building Block — A reusable, implementation-independent description of required capability or behavior. Compare with [Solution Building Block](#).

Architecture capability — The people, roles, skills, method, governance, information, tools, and organizational relationships needed to perform useful architecture work repeatedly.

Architecture change management — The monitoring and decision process used to keep architecture relevant as strategy, operations, technology, risk, and delivery evidence change. It is the purpose of ADM Phase H.

Architecture Change Request — A controlled request to evaluate a proposed change to architecture scope, content, requirements, standards, or an approved implementation.

Architecture compliance — The degree to which an implementation follows applicable architecture requirements, principles, standards, and decisions.

Architecture compliance review — A formal or proportionate review of implementation evidence against applicable architecture requirements, principles, standards, and decisions.

Architecture Continuum — The Enterprise Continuum dimension classifying architecture assets from foundational and common-system material through industry and organization-specific architectures.

Architecture Contract — An agreement between architecture and implementation stakeholders covering responsibilities, conformance evidence, review points, exceptions, and acceptance.

Architecture debt — The accumulated cost, risk, or constraint caused by architecture compromises, outdated decisions, duplication, or deferred remediation.

Architecture decision — A choice with material consequences for enterprise structure, behavior, standards, investment, risk, or future options.

Architecture Decision Record — A compact record of a decision's context, options, choice, rationale, consequences, owner, status, and revisit triggers.

Architecture Definition Document — The integrated architecture description, normally bringing together relevant baseline, target, gaps, rationale, and views across the selected domains.

Architecture description — The collection of work products used to express an architecture for its stakeholders and purposes.

Architecture Development Method — TOGAF's configurable, iterative method for moving from architecture capability and vision through domain development, migration planning, implementation governance, and change management.

Architecture domain — A major subject area of architecture work. The four commonly used TOGAF domains are Business, Data, Application, and Technology.

Architecture framework — A conceptual structure, method, vocabulary, content approach, and supporting practices used to develop and govern architectures.

Architecture governance — The system of decision rights, criteria, controls, accountabilities, conformance, exception handling, and feedback used to direct architecture.

Architecture Landscape — The organized view of baseline, transition, and target architectures across scopes, domains, levels, and time horizons.

Architecture level — The level of detail and decision horizon at which architecture is developed. Common examples include strategic, segment, capability, portfolio, product, and solution architecture.

Architecture maturity — The degree to which an architecture capability is established, repeatable, governed, measured, integrated, and continuously improved.

Architecture model — A purposeful representation of selected architecture concepts and relationships.

Architecture partitioning — Dividing a large architecture landscape or body of work into manageable, governed parts while maintaining boundaries, dependencies, ownership, and overall coherence.

Architecture pattern — A reusable response to a recurring architecture problem, including its context, forces, solution structure, and consequences.

Architecture principle — A durable rule or guideline used to make consistent decisions. A useful principle includes a statement, rationale, and implications.

Architecture Repository — The managed storage and operating mechanism for architecture content, standards, reusable assets, governance records, and capability information.

Architecture requirements repository — The controlled store of architecture requirements, attributes, owners, status, priorities, relationships, and traceability across ADM work.

Architecture Requirements Specification — The controlled set of measurable requirements that the architecture and its implementations must satisfy.

Architecture Roadmap — A strategic, time-oriented sequence of work packages and transition architectures showing dependencies and intended outcomes.

Architecture runway — Enabling architecture capability prepared ahead of near-term product or delivery needs. This is common Agile architecture language, not a mandatory TOGAF artifact.

Architecture Vision — A concise, stakeholder-oriented explanation of the drivers, scope, high-level baseline and target, value, risks, requirements, and proposed approach.

Artifact — A representation of architecture information. TOGAF commonly groups artifacts into catalogs, matrices, and diagrams.

B

Baseline Architecture — The relevant current state used as the starting point for analysis. It should be detailed only enough to support the decision.

Boundaryless Information Flow — A vision of integrated information being available to authorized users when and where needed across organizational, system, and technology boundaries.

Building block — A potentially reusable unit of architecture or solution content. Building blocks have responsibilities, interfaces, relationships, and required qualities.

Business Architecture — The domain covering strategy, capabilities, value streams, organization, actors, services, processes, information concepts, and operating model.

Business capability — An ability the enterprise possesses or needs to achieve an outcome. A capability describes **what** the enterprise can do, rather than a particular organization or process for doing it.

Business capability map — A structured view of business capabilities, usually organized into stable levels and used for investment, ownership, heat mapping, or change analysis.

Business driver — An internal or external condition creating a reason for action, such as regulation, competition, cost, customer expectations, or technology obsolescence.

Business footprint diagram — An artifact relating business goals, organizational units, functions, and services to show where change has impact.

Business goal — A broad intended result that gives direction to more specific objectives and measures.

Business interaction matrix — An artifact showing interactions, dependencies, or information exchange among business actors, roles, units, or functions.

Business outcome — A meaningful result or change in business performance, behavior, risk, or stakeholder value.

Business scenario — A structured description of a business problem, environment, actors, process, desired outcomes, measures, and implied requirements.

Business service — An externally or internally visible unit of business behavior that provides value to a consumer.

Business Transformation Readiness Assessment — An assessment of factors such as sponsorship, culture, skills, capacity, funding, governance, and stakeholder commitment that affect the ability to execute change.

C

Capability — An ability to achieve an effect or outcome. Capabilities can be business, architecture, organizational, solution, or technology capabilities.

Capability Architecture — Architecture focused on developing or changing a particular capability across the business, data, application, technology, and delivery concerns needed to realize it.

Capability-Based Planning — Planning that starts with the capabilities required to achieve strategic outcomes, then coordinates increments of change across organizational boundaries.

Catalog — An artifact that lists architecture entities of one kind, such as stakeholders, requirements, capabilities, applications, standards, or risks.

Change — A controlled movement from a baseline toward a target or transition state. Change can affect architecture, implementation, organization, or the architecture capability itself.

Communications plan — A plan mapping stakeholders and concerns to messages, views, channels, timing, ownership, and feedback.

Compliance assessment — Evidence-based evaluation of whether a solution or implementation conforms to applicable architecture requirements, principles, standards, and decisions.

Compliance review — The governance activity that performs or considers a compliance assessment and determines required actions or exceptions.

Concern — An interest, issue, need, risk, or expectation important to a stakeholder. Views should be selected to address defined concerns.

Conformance — Satisfaction of specified requirements, rules, standards, or contractual expectations.

Constraint — A limitation or condition that restricts the available architecture choices. Constraints may be legal, financial, organizational, technical, contractual, or time-based.

Content framework — The structured approach used to organize architecture deliverables, artifacts, and building blocks.

Content metamodel — The definition of architecture entity types, relationships, attributes, and rules that govern architecture content.

D

Data Architecture — The domain covering data meaning, structure, ownership, quality, use, movement, lifecycle, governance, and protection.

Data entity — A concept representing information of interest to the enterprise, such as Customer, Product, Contract, or Assessment.

Data product — A managed, discoverable data asset or service with defined consumers, ownership, quality expectations, interfaces, and lifecycle.

Decision rights — Explicit authority defining who proposes, advises, decides, approves exceptions, and is accountable for each type of decision.

Deliverable — A formally reviewed work product. A deliverable may contain multiple artifacts and may be contractually specified.

Dependency — A relationship in which one outcome, decision, work package, system, or capability relies on another.

Diagram — An artifact used to communicate structure, behavior, sequence, context, or relationships visually.

Digital enterprise — An organization whose strategy, operating model, products, decisions, and interactions substantially depend on digital capabilities and information.

E

Enterprise — The organizational scope addressed by the architecture. It can be an entire company, public body, partnership, business unit, ecosystem, or another purposeful boundary.

Enterprise Architecture — The discipline that connects strategic intent and desired outcomes to coordinated change across business, data, applications, technology, and cross-cutting concerns.

Enterprise Architecture capability — Another form of **architecture capability**, emphasizing its enterprise-wide mandate, relationships, and operating model.

Enterprise architecture levels — Related scopes and decision horizons at which architectures are developed. Organizations commonly distinguish strategic, segment, capability, portfolio, product, and solution levels.

Enterprise Continuum — A classification mechanism for understanding architecture and solution assets from generic to organization-specific. It is not the same as a physical repository.

Exception — A formally reviewed and authorized departure from an applicable architecture requirement, principle, standard, or decision. Good exceptions have scope, rationale, risk, owner, conditions, and expiry or review date.

F

Fitness for purpose — The degree to which architecture work is sufficient, usable, timely, and appropriate for the decision and risk at hand.

Foundation Architecture — A generic architecture of broadly applicable services, functions, and standards from which more specific architectures can be developed. This is prominent in classic TOGAF continuum material.

Foundation certification — The TOGAF Enterprise Architecture learning level focused on terminology, concepts, ADM structure, foundational techniques, governance, and content knowledge.

G

Gap — A meaningful difference between baseline and target that implies change, such as something to add, remove, replace, improve, retain, or govern differently.

Gap analysis — The structured comparison of baseline and target to identify required changes, dependencies, risks, and candidate work packages.

Governance Log — The repository area for decisions, compliance results, exceptions, risks, issues, actions, and other architecture governance evidence.

I

Implementation and Migration Plan — The prioritized, resourced, and integrated plan for executing work packages, managing dependencies, and reaching transition and target states.

Implementation Factor Assessment and Deduction Matrix — A structured technique for capturing factors that affect implementation and migration, assessing their impacts, and deriving actions, risks, or plan changes.

Implementation Governance — Oversight that helps implementation realize the approved architecture through clear agreements, review points, evidence, conformance, and managed deviations. It is the focus of ADM Phase G.

Information Systems Architectures — The collective term for Data Architecture and Application Architecture in ADM Phase C.

Interoperability — The ability of people, processes, information, and systems to exchange and use services or information effectively across boundaries.

Interoperability requirement — A condition defining the organizational, semantic, information, application, or technology compatibility required between parties or systems.

Iteration — Repeating or revisiting architecture work as understanding improves. TOGAF supports iteration within phases, between phases, across ADM cycles, and at different architecture levels.

M

Matrix — An artifact showing relationships between two or more kinds of architecture content, such as capability/application or stakeholder/concern.

Measure — A defined observation used to evaluate progress, performance, risk, conformance, or outcomes.

Metamodel — A model defining the concepts, relationships, attributes, and rules used to create other models.

Migration Planning — The prioritization and sequencing of work packages using value, cost, risk, dependency, readiness, and resource evidence. It is the focus of ADM Phase F.

Model — A simplified representation of a subject created for a defined purpose and audience.

O

Objective — A specific result to be achieved in support of a broader goal, ideally with a measure and time horizon.

Opportunities & Solutions — ADM Phase E, where architecture gaps are grouped into work packages, solution options are considered, and candidate transition architectures are formed.

Organization map — An artifact showing organizational units, roles, locations, responsibilities, reporting, or other relationships relevant to the architecture.

Organization model for Enterprise Architecture — The definition of architecture scope, roles, responsibilities, governance, skills, services, and connections to the rest of the organization.

Outcome — The effect or result produced for a stakeholder, as distinct from an output such as a document, system, or completed activity.

P

Pattern — A reusable response to a recurring problem in a stated context, including the forces, structure, and trade-offs involved.

Portfolio — A managed collection of investments, products, programs, projects, services, or assets considered together for prioritization and governance.

Practitioner certification — The TOGAF Enterprise Architecture learning level focused on analyzing and applying the framework in realistic situations.

Principle — A durable rule or guideline used to make consistent decisions. Architecture principles should have explicit rationale and implications.

Product — A persistently managed offering that provides value to customers or users and evolves through an ongoing lifecycle.

Project — A temporary endeavor organized to deliver a defined change or result.

R

Reference Architecture — A reusable architecture approach for a recurring class of problem, providing structures, patterns, decisions, and guidance that must be specialized for a context.

Reference Library — The Architecture Repository area containing reusable patterns, reference architectures, templates, viewpoints, examples, and lessons.

Reference Model — An abstract, reusable model defining key concepts, functions, services, or relationships for a domain.

Requirement — A statement of what an architecture or implementation must enable, satisfy, constrain, or demonstrate.

Requirements impact assessment — Analysis of how a new or changed requirement affects scope, stakeholders, architectures, gaps, work packages, plans, risk, and governance.

Requirements Management — The continuous ADM activity for eliciting, analyzing, prioritizing, tracing, verifying, and controlling requirements while feeding impacts into every phase.

Requirements repository — The controlled store for requirements and their attributes, status, owners, priorities, relationships, decisions, and traceability.

Risk — The effect of uncertainty on objectives. Architecture risk should record likelihood, impact, exposure, treatment, owner, evidence, and review.

Risk management — The continuous identification, assessment, treatment, monitoring, and communication of uncertainty affecting architecture and transformation outcomes.

S

Scope — The boundaries of architecture work, commonly considered by breadth, depth, time period, architecture domains, organizational coverage, and subject matter.

Security Architecture — Architecture that expresses security outcomes, threats, controls, trust boundaries, identity, protection, detection, response, and recovery across all domains.

Segment Architecture — Architecture for a coherent part of an enterprise, such as a business segment, mission area, region, or major portfolio.

Service — Defined behavior or capability offered by one party or component to another through an understood interface or agreement.

Solution Architecture — The architecture of a bounded initiative or solution, connecting enterprise and domain direction to implementable choices.

Solution Building Block — A specific implementable or procured component that realizes one or more Architecture Building Blocks.

Solution concept diagram — A high-level visual summary of the major solution participants, services, interactions, and technology direction used to communicate an emerging solution concept.

Solutions Continuum — The Enterprise Continuum dimension classifying implemented solutions from generic products and services through industry-specific and organization-specific solutions.

Stakeholder — A person, group, role, or organization with an interest in or influence over the architecture and its outcomes.

Stakeholder management — The continuous identification and analysis of stakeholders, concerns, influence, decision rights, engagement, communications, and changing support.

Standard — An approved rule, specification, product choice, interface, technology, or practice that constrains or guides implementation in a defined scope.

Standards Information Base — The Architecture Repository area containing standards, approved technologies and products, versions, lifecycle status, owners, exceptions, and migration guidance.

Statement of Architecture Work — The agreement defining an architecture engagement's objectives, scope, approach, work plan, roles, governance, deliverables, acceptance criteria, assumptions, constraints, and risks.

Strategic Architecture — A high-level, longer-horizon architecture that guides major enterprise direction and provides context for more focused segment, capability, product, or solution work.

T

Tailoring — Deliberately configuring the ADM, content, techniques, governance, depth, cadence, and deliverables to suit outcomes, scope, risk, uncertainty, culture, and delivery context.

Target Architecture — The intended future state designed to satisfy agreed drivers, goals, stakeholder concerns, and requirements.

Technology Architecture — The domain covering platforms, infrastructure, networks, runtime services, middleware, technology standards, operations, and lifecycle.

TOGAF Fundamental Content — The stable core of the modular 10th Edition, including concepts, the ADM, techniques, applying the ADM, content, capability, and governance.

TOGAF Series Guide — Topic-focused guidance for configuring or applying the TOGAF framework in a particular area. Series Guides can evolve more frequently than the Fundamental Content.

Traceability — The ability to follow relationships from drivers and stakeholder concerns through goals, requirements, architecture decisions, gaps, work packages, implementation, and measured outcomes.

Trade-off analysis — A structured comparison of credible options against stakeholder concerns, criteria, constraints, uncertainty, risk, cost, value, reversibility, and consequences.

Transition Architecture — A coherent intermediate architecture state between baseline and target, useful when the target cannot be reached in one step.

V

Value — The benefit, usefulness, or importance perceived by a stakeholder relative to cost, risk, sacrifice, or alternatives.

Value stream — The end-to-end stages through which value is created and delivered to a stakeholder.

View — The actual representation of an architecture created from a viewpoint to address particular stakeholder concerns.

Viewpoint — The conventions, concerns, model kinds, and rules used to construct, interpret, and use a view.

W

Waiver — A common synonym for an approved exception. It should be governed, bounded, owned, and time-limited rather than treated as permanent silent non-conformance.

Work package — A bounded set of coordinated actions or change used to realize part of an Architecture Roadmap.

Architecture Repository areas

Repository area	What belongs there
Architecture Metamodel	Content types, relationships, attributes, rules, and modeling conventions
Architecture Capability	Roles, skills, organization, governance, tailored method, services, and tools
Architecture Landscape	Baseline, transition, and target architecture descriptions across scope and time
Standards Information Base	Standards, approved products and versions, lifecycle status, ownership, and exceptions
Reference Library	Patterns, reference architectures, templates, viewpoints, examples, and lessons
Governance Log	Decisions, compliance findings, exceptions, risks, actions, and architecture debt

Deliverables and work products

Term	Primary purpose
Request for Architecture Work	Formally trigger and frame a need for architecture work
Architecture Vision	Explain the proposed change, scope, value, target direction, stakeholders, and risks
Statement of Architecture Work	Authorize and control the architecture engagement
Architecture Definition Document	Describe the baseline, target, gaps, rationale, and relevant views
Architecture Requirements Specification	State the measurable conditions the architecture and implementation must satisfy
Architecture Roadmap	Show the strategic sequence of work packages and transition states
Implementation and Migration Plan	Turn the roadmap into a prioritized, resourced delivery plan
Architecture Contract	Agree responsibilities, conformance, evidence, review points, and exceptions
Compliance Assessment	Record evidence of conformance or deviation
Architecture Change Request	Request controlled evaluation of a change

Commonly confused terminology

Pair	Fast distinction
TOGAF / Enterprise Architecture	TOGAF is a framework; Enterprise Architecture is the discipline and resulting architecture
ADM / project lifecycle	ADM develops and governs architecture; a project lifecycle manages delivery
Stakeholder / concern	A stakeholder has an interest; a concern is what matters to that stakeholder
View / viewpoint	A view is the representation; a viewpoint defines how it is constructed and used
Model / view	A model represents selected concepts; a view presents relevant model content for concerns
Baseline / target	Relevant current state / intended future state
Gap / requirement	Difference implying change / condition to be satisfied
Transition Architecture / target	An intermediate coherent state / the intended end state for the defined horizon
Transition Architecture / roadmap	An intermediate state / the sequence for reaching states
Roadmap / Implementation and Migration Plan	Strategic sequence / executable, resourced plan
Deliverable / artifact	Formally reviewed package / representation of architecture information
Catalog / matrix / diagram	List / relationships / visual structure or behavior
ABB / SBB	Required capability or behavior / specific implementation realization
Architecture Continuum / Solutions Continuum	Classification of architectures / classification of implementations
Enterprise Continuum / Architecture Repository	Classification mechanism / managed storage and control mechanism
Principle / requirement	Durable decision rule / specific condition to satisfy
Standard / pattern	Approved rule or choice / reusable response with contextual trade-offs
Compliance / governance	Evidence of conformance / the broader system of decisions, controls, exceptions, and feedback

Pair	Fast distinction
Capability / process	Ability to achieve an effect / sequence of activities used to operate
Outcome / output	Result or effect / thing produced
Product / project	Persistent value-delivering offering / temporary change endeavor

See also

- [Fast-learning mind map](#)
- [Editable draw.io mind map](#)
- [Core concepts](#)
- [Architecture Development Method](#)
- [Content, deliverables, and artifacts](#)
- [Enterprise Continuum and Architecture Repository](#)
- [Capability and governance](#)
- [Official references](#)

This glossary is an independent learning aid. The [TOGAF Standard publications catalog](#) and [TOGAF Library](#) remain the authoritative sources for current standard material.

Final TOGAF Readiness Assessment

Use this only after studying the handbook. It answers one question:

Can I explain TOGAF, apply it to a situation, and perform under exam conditions?

The interactive version is in the [TOGAF Study Desk](#). Progress is stored only in your browser.

The five gates

All five gates are required. A high score in one area does not hide a weakness in another.

Gate	Evidence needed	Pass rule
Recall	Spaced-repetition flashcards	At least 70% of the 72 cards mastered
Coverage	Explain every published syllabus area without notes	14 of 14 areas checked
Connected understanding	Twelve-question diagnostic	At least 80%
Application	Teach, draw, and solve cases	All three evidence tasks complete
Timed performance	Full mock exams	Two Foundation and two Practitioner passes

Evidence tasks

1. Teach it

Explain TOGAF aloud in two minutes without notes. Include:

- the problem TOGAF solves;
- baseline, target, gaps, and roadmap;
- Preliminary, Phases A–H, and Requirements Management;
- Business, Data, Application, and Technology;
- stakeholders, governance, implementation, and change.

If a listener cannot repeat the big idea back to you, simplify and try again.

2. Draw it

On blank paper, draw:

- Preliminary before the cycle;
- Phases A through H in order;
- Requirements Management interacting with every phase;
- the four domains;
- a feedback arrow from real-world change back into architecture work.

Then explain the drawing without reading labels.

3. Apply it

Complete all eight cases in the [Practitioner case-study workbook](#). Use its rubric. Aim for at least 3 out of 4 on every case.

4. Prove it under time

Use different sets in the [mock exam simulator](#):

1. pass Foundation twice at 24/40 or better;
2. pass Practitioner twice at 24/40 points or better;
3. do not count two immediate retries of the same memorized set;
4. repair weak topics between attempts;
5. finish the final attempt with at least five minutes available.

Final oral check

Answer each question in one or two simple sentences:

1. What is TOGAF trying to make easier?
2. Why is the ADM not a waterfall?
3. What does each ADM phase decide?
4. Why is Requirements Management central?
5. How are stakeholder, concern, viewpoint, and view connected?
6. What changes between baseline and target?
7. How do Phases E, F, and G differ?
8. What is the difference between a deliverable, artifact, and building block?
9. How does governance handle a deviation?
10. How does Phase H decide whether to maintain, iterate, or start a new cycle?

Any answer that depends on jargon you cannot explain is not yet secure. Return to the smallest relevant handbook section, explain it in plain language, and test yourself again.

Ready means

You are ready when you can:

- recall the framework without cues;
- choose a proportionate action in an unfamiliar scenario;
- explain why better options are better;
- find supporting material quickly in an open-book setting; and
- repeat a passing result under time.

This assessment is an independent learning aid. It is not an official result or a guarantee of certification performance.