Zachman: The Rationale for Enterprise Architecture and the Zachman Framework

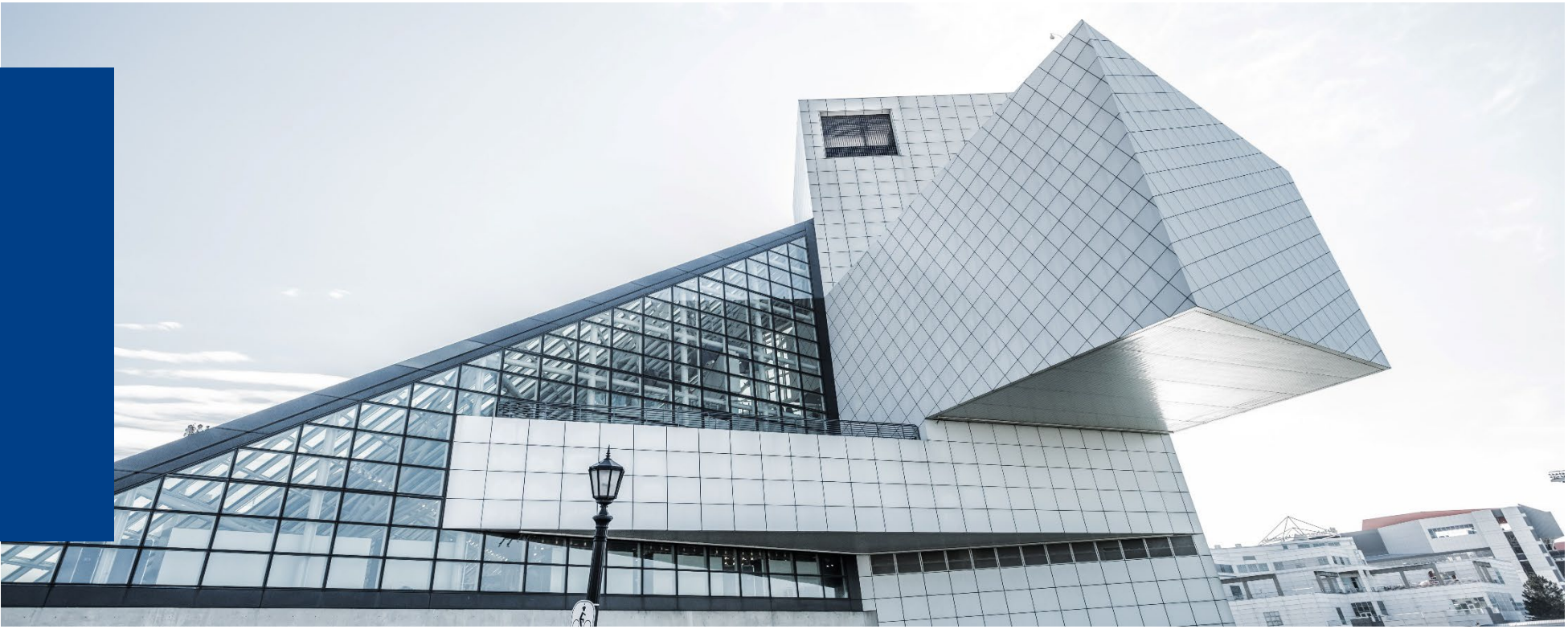


Photo by Lance Anderson on Unsplash

- Based on the Lecture of John Zachman, October 2015

The Zachman Framework

Zachman Framework

- Regarded the origin of enterprise architecture frameworks (originally called "Framework for Information Systems Architecture")
- First version published in 1987 by John Zachman
- It is still further developed by Zachman International (http://www.zachman.com)
- Often referenced as a standard approach for expressing the basic elements of enterprise architecture

slido

We need Enterprise Architecture because ...

ⓘ Start presenting to display the poll results on this slide.

In short, the reasons you need Architecture:

COMPLEXITY AND CHANGE

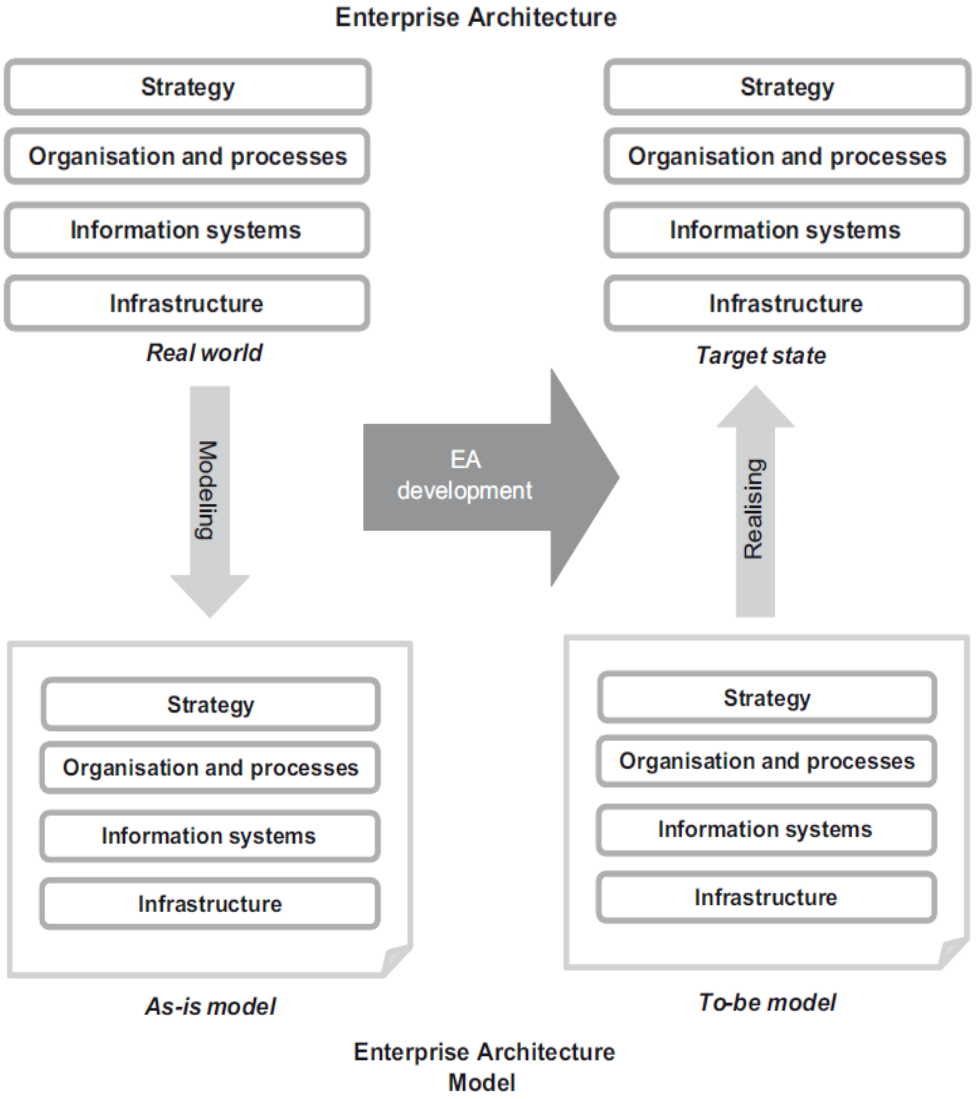
Architecture: Dealing with Complexity and Change

- If the object you want to create or change is simple, and it is not likely to change, then you can do it directly.
- On the other hand, if the object is **complex**, you can't see it in its entirety at one time and it is likely to **change** considerably over time, you need a description or model.
- You need a description of the **Architecture**.

Why Enterprise Architecture Description?

- **Manage change of complex system**
  - Baseline for complex, interdependent enterprise decisions
  - Communication of decisions to organization stakeholders.
  - *If architecture is not explicit, there is a high risk that the implementation is not what is intended*
- **Continuous, coordinated organisation change**
  - Continuously update Enterprise Architecture to reflect changes
  - Coordinate change between different projects

Change the Model before you Change the System!



slido

What is reification?

ⓘ Start presenting to display the poll results on this slide.

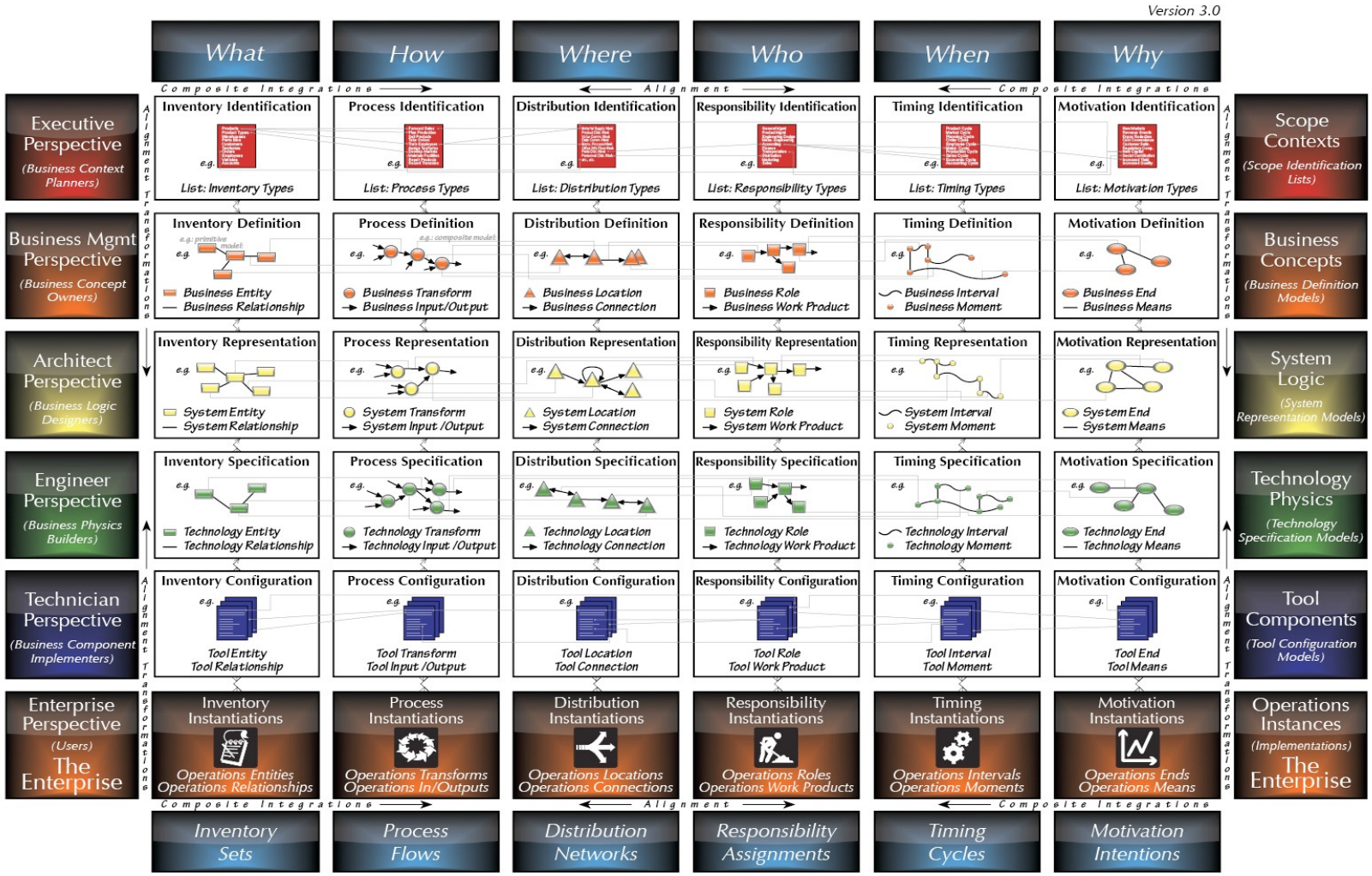
Reification

- Reification is the process by which an abstract idea is turned into an explicit object or thing

The Zachman Framework

Abstractions

Perspectives

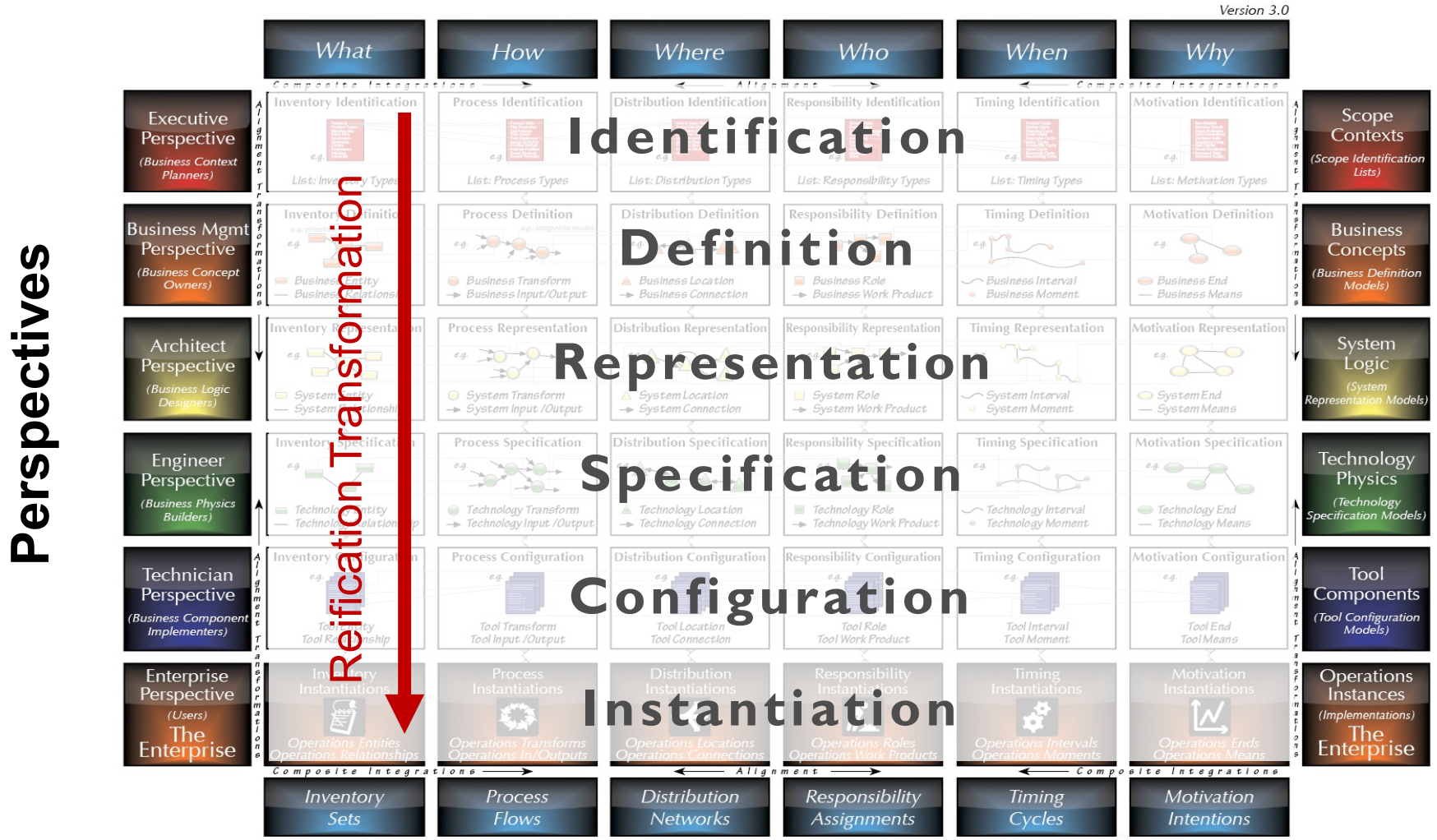


slido

To which dimension of the Zachman Framework does Reification correspond?

ⓘ Start presenting to display the poll results on this slide.

Dimension 1: Reification
Turning an abstract Idea into something Concrete



Dimension 1 – Perspectives

Zachman originally used the analogy of classical architecture

From the idea to the building

On each level different aspects of a building are relevant - models of the building from different perspectives

Bubble charts: conceptual representation delivered by the architect

Architect's drawing: transcription of the owner's perceptual requirements – *owner's perspective*

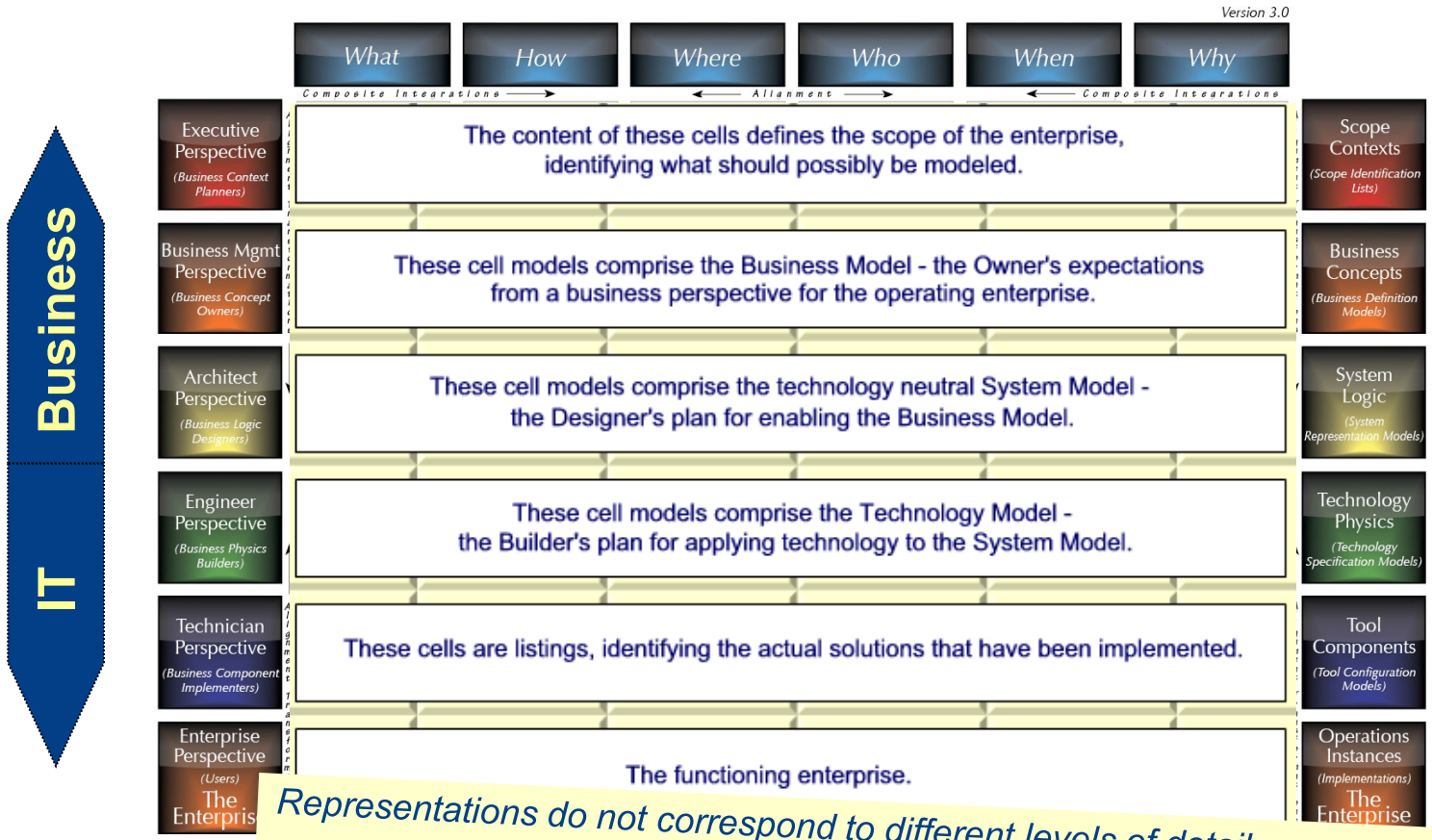
Architect's plans: translation of the owner's requirements into a product – *designer's perspective*

Contractor's plans: phases of operation, architect's plans contrained by nature and technology – *builder's perspective*

Shop plans: parts/sections/components of building details (out-of-context specification) – *subcontractor's perspective*

The building: physical building itself

Reification: Turning an Abstract Idea (Strategy) into Something Concrete (Functioning Enterprise)



Scope (Boundaries)

Requirements (Concepts)

Design (Logic)

Plan (Physics)

Part (Configurations)

Product (Instantiations)

Representations do not correspond to different levels of details

- *Each row is different in nature, in content, in semantics from the others – representing different perspectives*
- *Level of detail is an independent variable, varying within one representation*

Dimension 1: Architectural Representations with analogies in Building and Information Systems

Generic	Buildings	Airplanes	Information Systems
Ballpark	Bubble charts	Concepts	Scope/objectives
Owner's representation	Architect's drawings	Work breakdown structure	Model of the business (or business description)
Designer's representation	Architect's plans	Engineering design/bill-of-materials	Model of the information system (or information system description)
Builder's representation	Contractor's plans	Manufacturing engineering design/bill-of-materials	Technology model (or technology-constrained description)
Out-of-context representation	Shop plans	Assembly/fabrication drawings	Detailed description
Machine language representation	—	Numerical code programs	Machine language description (or object code)
Product	Building	Airplane	Information system

slido

What are the six abstractions of the Zachman Framework?

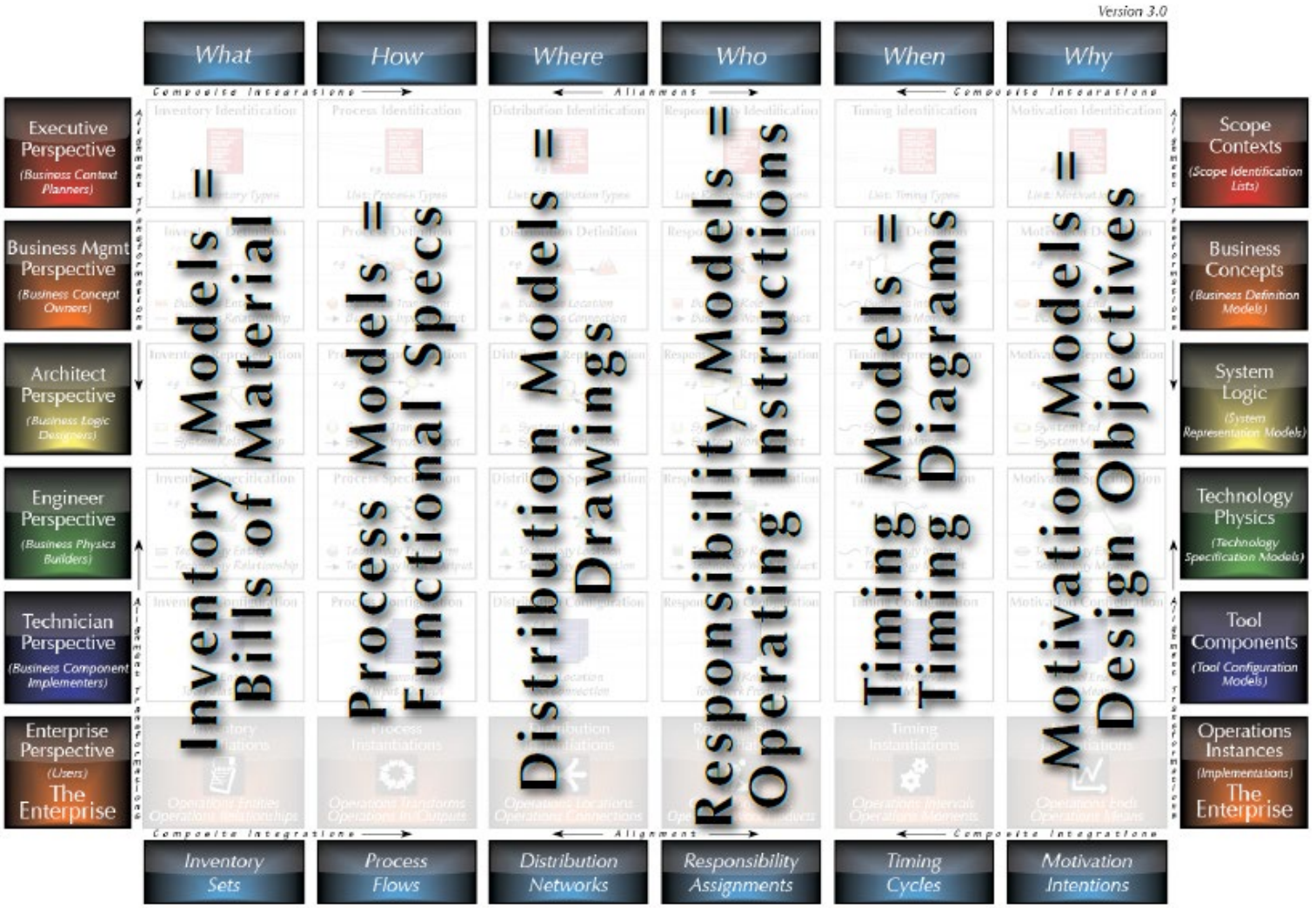
ⓘ Start presenting to display the poll results on this slide.

Dimension 2: Aspects of an Architecture

- There exist different types of descriptions oriented to different aspects
- Zachman associates each aspect with a question word

WHAT	inventory models
HOW	functional/process models
WHERE	location/distribution models
WHO	organisation models
WHEN	timing models
WHY	motivation models

Abstractions for Manufacturing



slido

What does quality mean in the context of the enterprise (architecture)?

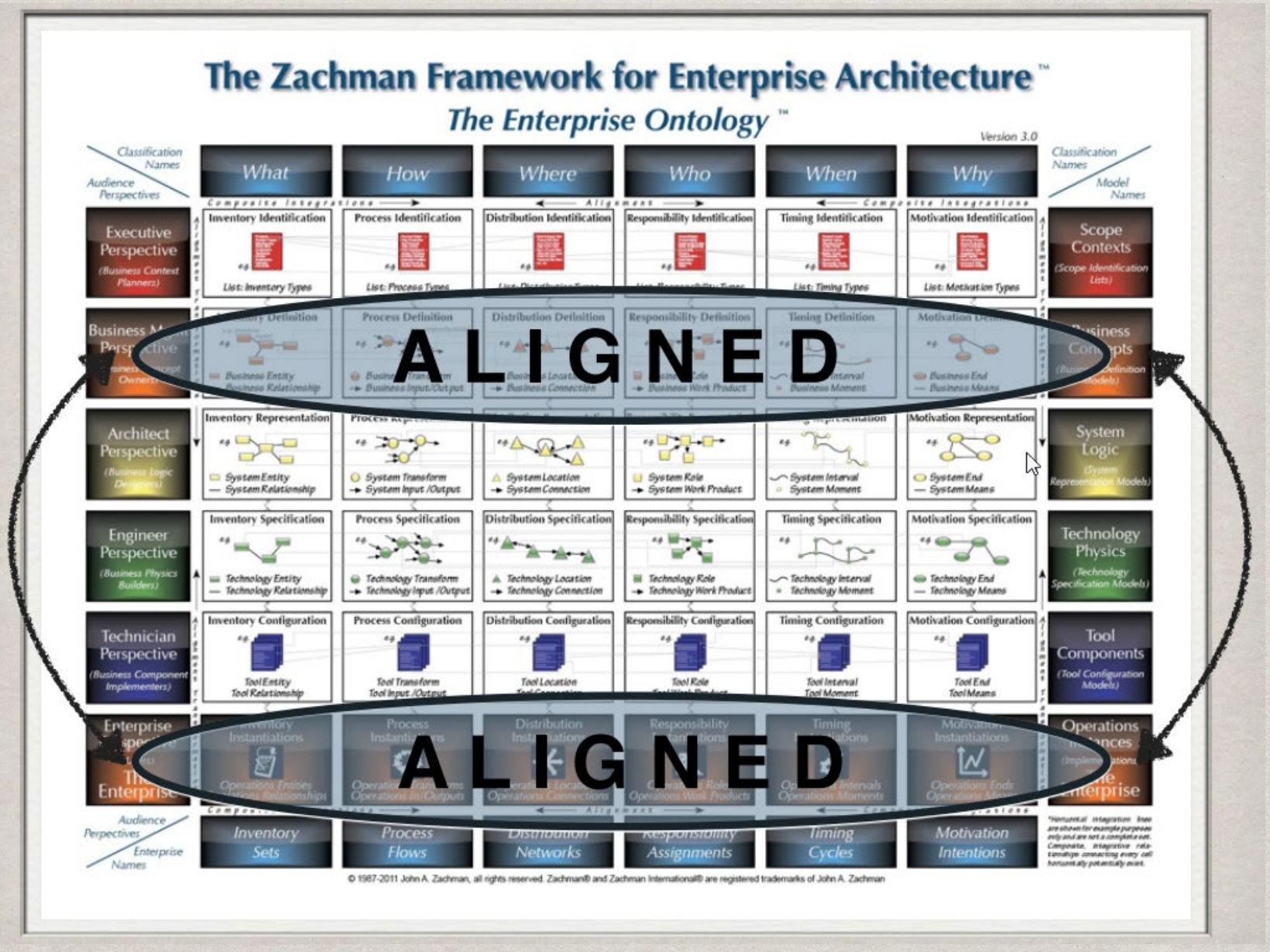
ⓘ Start presenting to display the poll results on this slide.

Quality

"Producing end results (the product)
that meet the requirements
as defined by the customer."

Quality in the context of the Enterprise

Producing Implementations
(manual and/or automated)
i.e. the ENTERPRISE (Row 6)
that are "aligned" with
the intentions of Management (Row 2).


The Zachman Framework for Enterprise Architecture™
The Enterprise Ontology™
Version 3.0
Classification Names
Audience Perspectives
What
How
Where
Who
When
Why
Model Names
Executive Perspective
(Business Context Planners)
Architect Perspective
(Business Logic Designers)
Engineer Perspective
(Business Physics Builders)
Technician Perspective
(Business Component Implementers)
Enterprise Perspective
Inventory Identification
Process Identification
Distribution Identification
Responsibility Identification
Timing Identification
Motivation Identification
List: Inventory Types
List: Process Types
List: Timing Types
List: Motivation Types
Process Definition
Distribution Definition
Responsibility Definition
Timing Definition
Business Entity
Business Relationship
Business Input/Output
Business Connection
Business Work Product
Business Moment
Business End
Business Means
Inventory Representation
Motivation Representation
System Entity
System Relationship
System Transform
System Input /Output
System Location
System Connection
System Role
System Work Product
System Interval
System Moment
System End
System Means
Inventory Specification
Process Specification
Distribution Specification
Responsibility Specification
Timing Specification
Motivation Specification
Technology Entity
Technology Relationship
Technology Transform
Technology Input /Output
Technology Location
Technology Connection
Technology Role
Technology Work Product
Technology Interval
Technology Moment
Technology End
Technology Means
Inventory Configuration
Process Configuration
Distribution Configuration
Responsibility Configuration
Timing Configuration
Motivation Configuration
Tool Entity
Tool Relationship
Tool Transform
Tool Input /Output
Tool Location
Tool Role
Tool Interval
Tool Moment
Tool End
Tool Means
Inventory Instantiations
Process Instantiations
Distribution Instantiations
Responsibility Instantiations
Timing Instantiations
Motivation Instantiations
Operations Entities
Operations Relationships
Operations Transforms
Operations Inputs/Outputs
Operations Locations
Operations Connections
Operations Roles
Operations Work Products
Operations Intervals
Operations Moments
Operations Ends
Operations Means
Scope Contexts
(Scope Identification Lists)
System Logic
(System Representation Models)
Technology Physics
(Technology Specification Models)
Tool Components
(Tool Configuration Models)
Operations Instances
(Implementations)
The Enterprise
Enterprise Names
Inventory Sets
Process Flows
Distribution Networks
Responsibility Assignments
Timing Cycles
Motivation Intentions
Composite Integrations
Alignment
ALIGNED
ALIGNED
© 1987-2011 John A. Zachman, all rights reserved. Zachman® and Zachman International® are registered trademarks of John A. Zachman

Quality Problems

Either:

A. The Requirements at Row 2 were incorrectly transcribed

Or:

B. In the transformation from Row 2 to Row 6, integrity was lost.

Or:

C. Whoever entered the data at Row 6 created errors.
(This is a Management problem, not an Architecture problem.)

Fixing Quality Problems

A. Fix the Process of transcribing the Requirements (Row 2)

And/Or:

B. Fix the Process of transforming the Requirements (Row 2) into Implementation (Row 6)

and Iterate
until Row 6 is aligned with Row 2.

CONTINUOUS PROCESS IMPROVEMENT

slido

Why does an enterprise need information systems people?

ⓘ Start presenting to display the poll results on this slide.

Enteprises need Information Systems People

- Information Systems people …
  - … bring to the table drafting skills
  - … can describe things very precisely
  - … can build models, which are unambiguous

- Is this important?
  - ♦ Yes! If several people are involved in a change process, they better should be able to look at the models and know precisely without ambiguity what is described

Modeling Skills

- Modeling skills consist of two aspects
  - Learning modeling languages
    - Syntax/Notation
    - Semantics
  - Being able to express reality appropriately
    - Pragmatics

Models and the Zachman Framework

- Concepts for modelling are related to cells.
- Cells shall contain models with concepts from a single abstraction perspective

slido

What statements about Ontology and Methodology are true?

ⓘ Start presenting to display the poll results on this slide.

Ontology

The Zachman Framework™ schema technically is an ontology -
a theory of the existence of a structured set
of essential components of an object

A Framework is a STRUCTURE.
(A Structure DEFINES something.)

Methodology

A Methodology is a PROCESS.
(A Process TRANSFORMS something.)

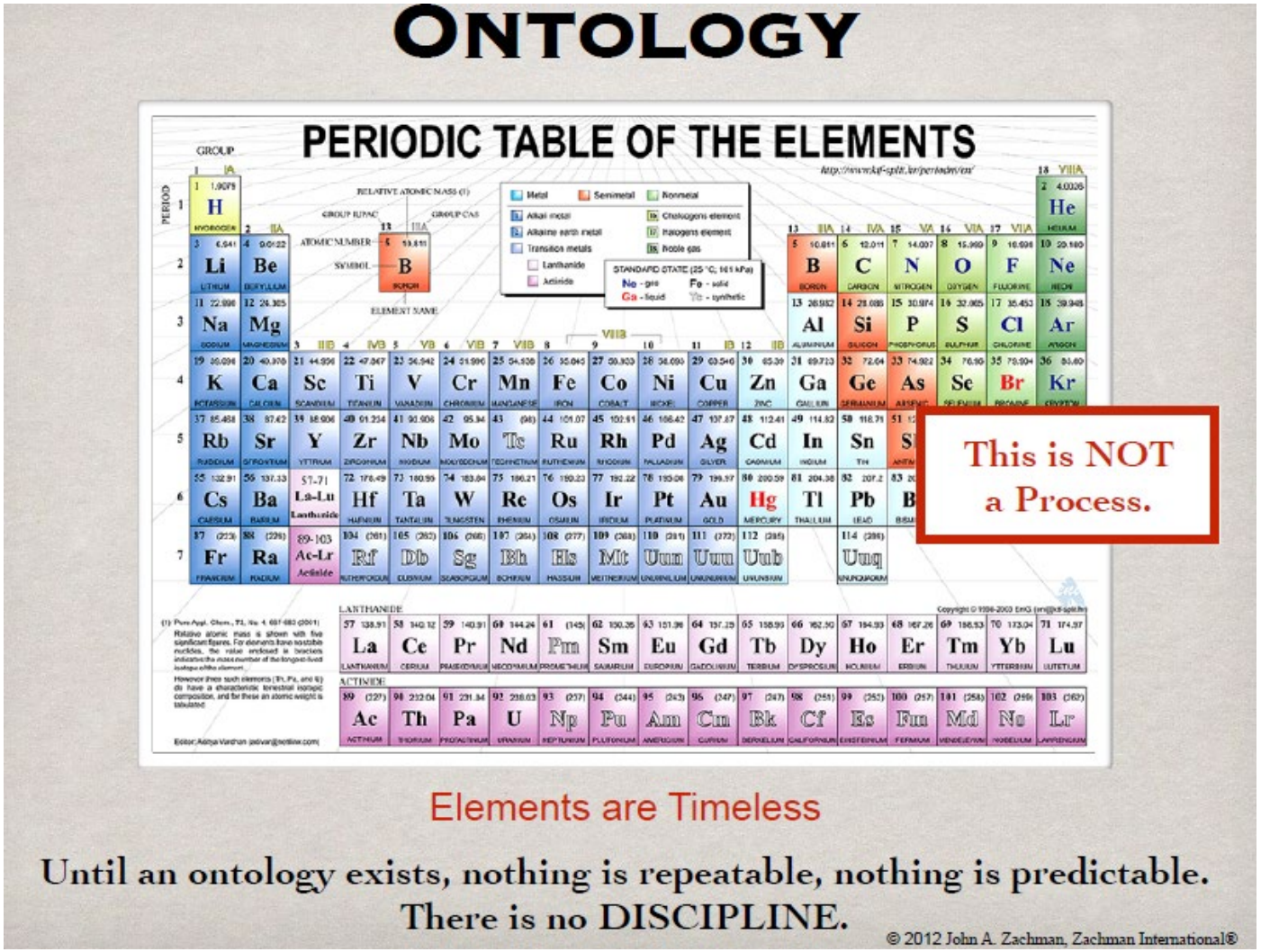
A Structure IS NOT A Process
A Process IS NOT a Structure.

Ontology vs Methodology

An Ontology is the classification of the total set of "**Primitive**" (elemental) components that exist and that are relevant to the existence of an object.

A Methodology produces "**Composite**" (compound) implementations of the Primitives.

Analogy: Chemistry



Analogy: Chemistry

PROCESS

(METHODOLOGY)

Add Bleach to an Alkali and
it is transformed into Saltwater.

$$HCl + NaOH \rightarrow NaCl + H_2O$$

COMPOUNDS

Salt	NaCl
Aspirin	$C_9H_8O_4$
Vicodin	$C_{18}H_{21}NO_3$
Naproxen	$C_{14}H_{14}O_3$
Ibuprophen	$C_{13}H_{18}O_2$
Viagra	$C_{22}H_{30}N_6O_4S$
Sulphuric Acid	H_2SO_4
Water	H_2O

etc., etc., etc.

This is NOT an Ontology.

Relations between Models and Model Elements

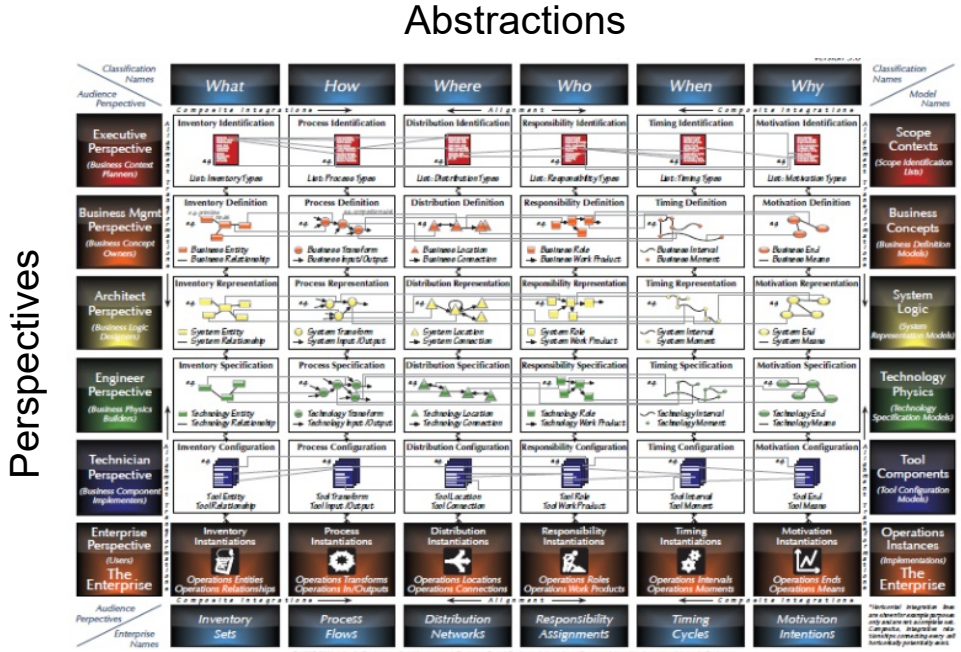
- There are relations between (elements of) the models
- Horizontal Relations: In same perspective, e.g.
  - Data used in a process
  - Application implementing a process activitiy
- Vertical relations: Between different perspectives
  - Implementation of an application
  - Database model for an entity relationship model

slido

Is the Zachman Framework an Ontology or a Methodolody or both?

ⓘ Start presenting to display the poll results on this slide.

The Zachman Framework



- The Zachman Framework is depicted as a 6 x 6 “matrix”.
  - ♦ Perspectives: Reification
  - ♦ Abstractions: Interrogatives
- The matrix constitutes the total set of descriptive representations that are relevant for describing anything, e.g. an enterprise → ONTOLOGY
- The Zachman Framework does not specify a methodology. Even for the Reification no process described, how to do it.

slido

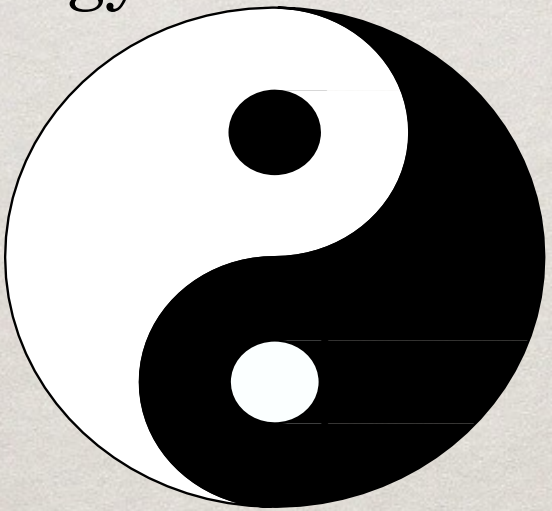
What is more important - Ontology or Methodology?

ⓘ Start presenting to display the poll results on this slide.

Ontology and Methodology

It is NOT either Ontology OR Methodology

It IS Ontology AND Methodology



Ontology and Methodologies
do not COMPETE they
COMPLETE

Alchemy - A Practice

This is a Methodology WITHOUT an Ontology

A Process with no ontological structure is ad hoc, fixed and dependent on practitioner skills.

This is NOT a science.

It is ALCHEMY,

a "**practice**."

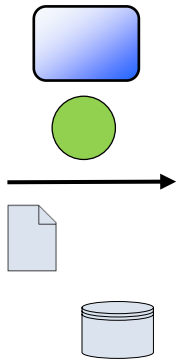
The Periodic Table Metaphor

Before Mendeleev published the Periodic table, Alchemist (practitioners) could create compounds based on their experience ... whatever worked. After Mendeleev figured out the Periodic Table, Chemistry became a science. Creating compounds became predictable and repeatable based on the natural laws (Physics) expressed in the Periodic Table. Within 50 years, the Chemists and Physicists (practitioners) were splitting atoms.

Ontology and Methodology in Business Process Management

Ontology

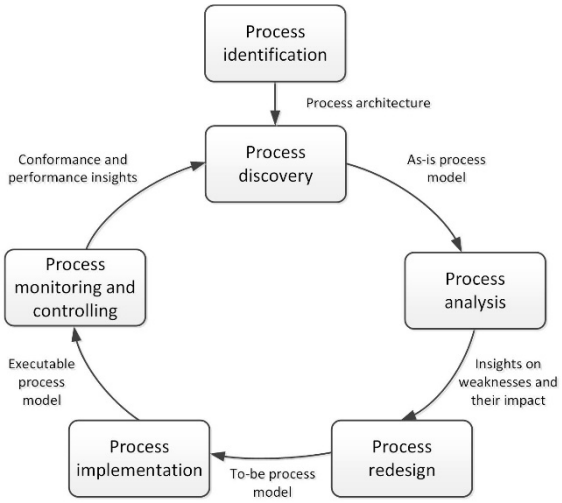
- Primitives for describing a business process
  - Activity
  - Event
  - Flow
  - Role
  - KPI
  - …



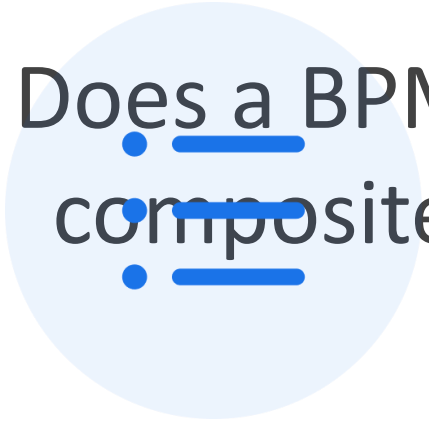
- Corresponds to elements and attributes of process modeling and their relations

Methodology

- The procedure for managing buiness processes: design, monitor, improve
- Corresponds to the BPM life cycle



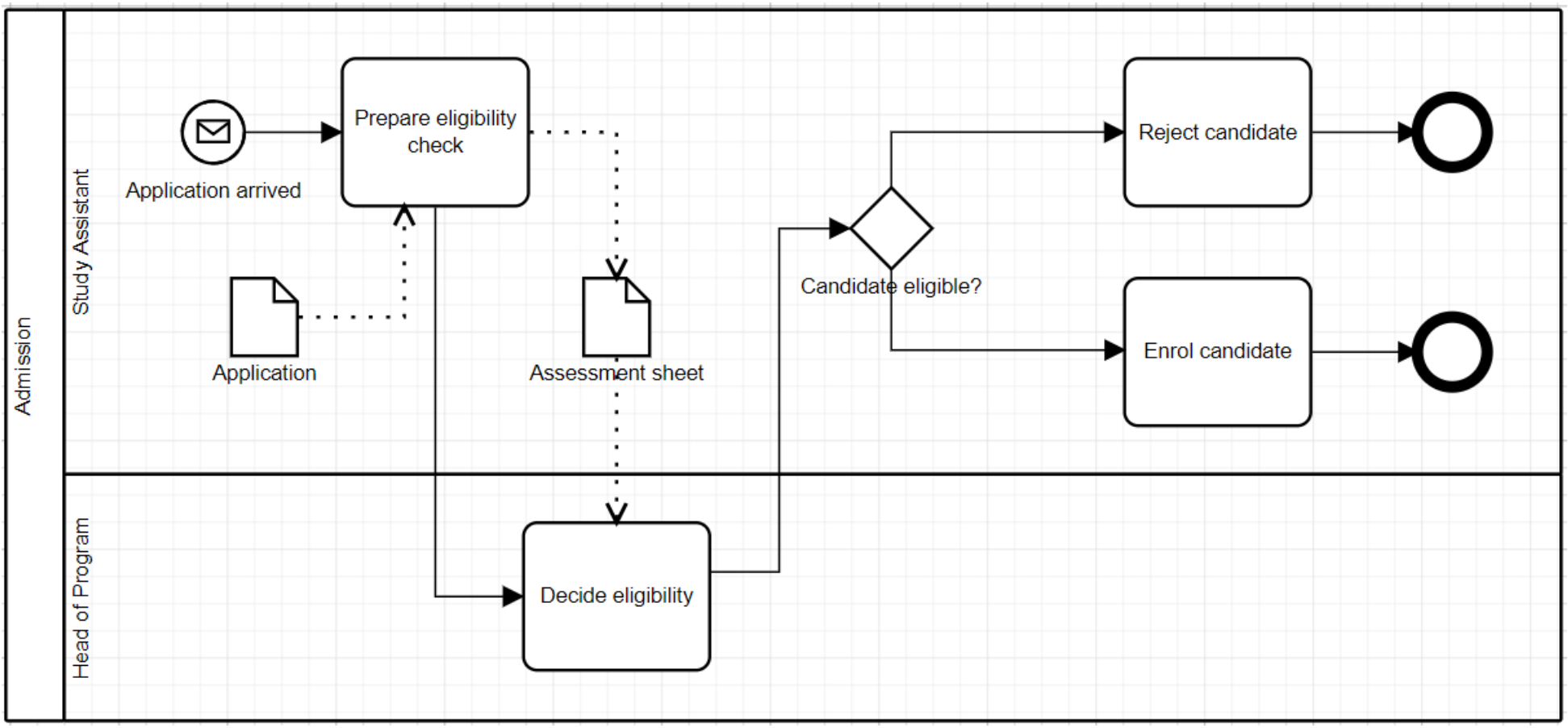
slido



Does a BPMN Model belong to one cell or is it a composite with elements of cells in different columns?

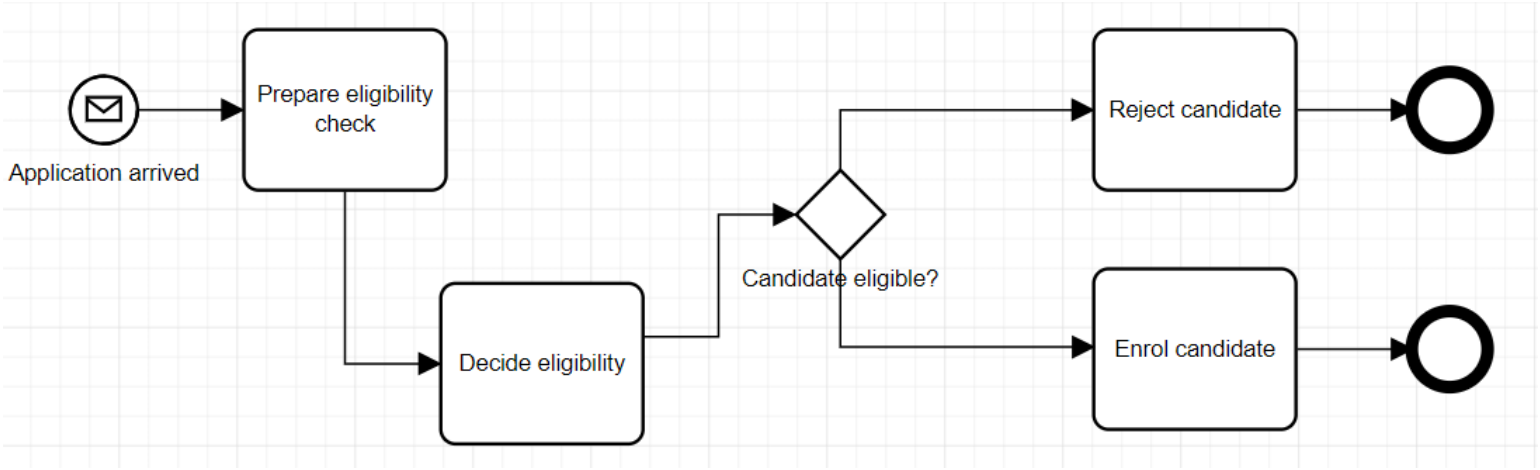
ⓘ Start presenting to display the poll results on this slide.

Composite

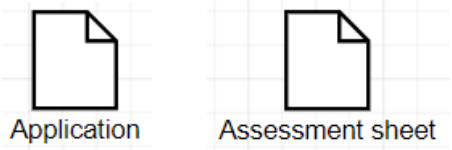

Admission
Study Assistant
Head of Program
Application arrived
Prepare eligibility check
Application
Assessment sheet
Candidate eligible?
Reject candidate
Enrol candidate
Decide eligibility

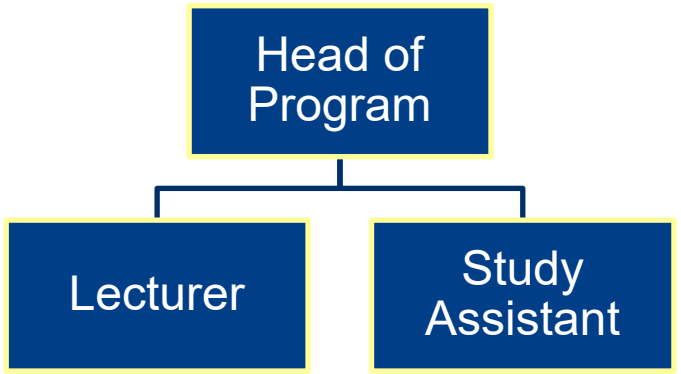
Primitives

- How



- What





Examples of Composites (COMPOUNDS)

- Java Programs
- Objects
- BPMN Models
- Business Architecture
- Capabilities
- Applications Data Models
- Security Architecture
- Services
- Technology Architecture
- Missions/Visions
- Agile Code
- Business Processes
- DoDAF Models
- Balanced Scorecard Clouds
- Archimate Artifacts

slido

According to Zachman's "Enterprise Physics": What is the case, if a cell of the Enterprise Ontology is not made explicit?

ⓘ Start presenting to display the poll results on this slide.

Why making Enterprise models

Every Cell of the Enterprise Ontology Exists

Any Cell that is not made explicit is ***implicit***, which means that you are allowing anyone and everyone to make whatever ***assumptions***

Incorrect assumptions are sources of ***defects*** … and the source of miscommunication and ***misunderstanding***

To avoid misunderstanding and miscommunication about the Enterprise, there should be only a ***single version of Cells in Rows 1, 2 and 3***

Enterprise Physics

The First Law of Enterprise Ontological Holism

Every Cell of the Enterprise Ontology exists. Any Cell or portion of Cell that is not made **explicit** is **implicit** which means that you are allowing anyone and everyone to make whatever assumptions they want to make about the contents and structure of that Cell.

The Second Law of Enterprise Ontological Holism

Correct assumptions about **implicit** Cell contents and structure save time and money. Incorrect assumptions are sources of defects … and the source of miscommunication and misunderstanding - conflicts, escalating General and Administrative costs (entropy) in the implemented Enterprise of Row 6.

Enterprise Physics

The Third Law of Enterprise, Ontological Holism. Every Cell or portion of Cell that is **not** explicit (i.e. is **im**plicit) is guaranteed to be a source of inconsistent assumptions and therefore discontinuities, risking potential conflicts, escalating General and Administrative costs (entropy) and even Enterprise liabilities.

The Fourth Law of Enterprise, Ontological Holism. To avoid misunderstanding and miscommunication about the Enterprise, there should be only a single version of Cells in Rows 1, 2 and 3. However, the Row 3 System Logic can be transformed to more than one Technology and the Row 4 Technology Physics transformed with more than one Vendor Tool as long as content redundancy is controlled.

Enterprise Physics

The Fifth Law of Enterprise, Ontological Holism.

Any fact that is not classifiable according to the defined classification rules is either not relevant to the Enterprise or not a single-variable, "Primitive" fact.

That fact (if it is a fact and if it is relevant to the Enterprise) is likely a "Composite" fact.

slido

The probability of the implemented Enterprise (Row 6) having anything to do with the intentions of the "stakeholders" of Rows 1, 2 or 3 is low to zero, if ...

ⓘ Start presenting to display the poll results on this slide.

The probability of the implemented Enterprise (Row 6) having anything to do with the intentions of the “stakeholders” of Rows 1, 2 or 3 is low to zero, if ...

… cells in ***Rows 1, 2 or 3 are not made explicit***

or

… cells in ***Rows 4 or 5 are not made explicit*** and *aligned with the Rows 1, 2 and 3*

Enterprise Physics

The First Law of Reification Incontrovertibility.

If Cells in Rows 1, 2 or 3 are not made explicit, whoever is formalizing Cells in Rows 4, 5 and 6 has to make assumptions about Rows 1, 2 and 3 and the probability of the implemented Enterprise of Row 6 having anything to do with the intentions of Rows 1, 2 or 3 is low to zero.

The Second Law of Reification Incontrovertibility.

If Cells in Rows 4, 5 or 6 are not made explicit and aligned with the transformations of Rows 1, 2 and 3, whether the Cells in Rows 1, 2 and 3 are made explicit and aligned or not, the probability of the implemented Enterprise of Row 6 having anything to do with the intentions of the "stakeholders" of Rows 1, 2 or 3 is low to zero.

slido

Is building models the purpose of Enterprise Architecture?

ⓘ Start presenting to display the poll results on this slide.

Challenge to Enterprise Architects

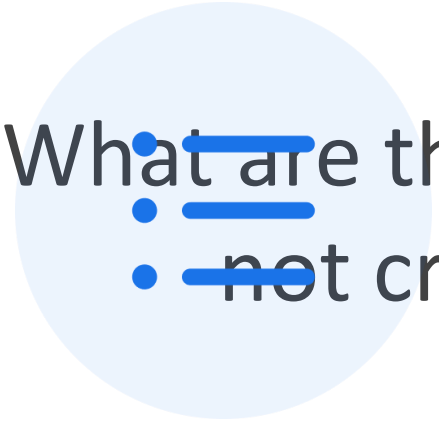
Reframe the concept of Enterprise Architecture ...

It is not about building models!

It is about solving Enterprise problems while iteratively and incrementally building out the inventory of complete, reusable, Primitive Models that constitute:

Enterprise Architecture.

slido



What are the options, if you do not retain (or did not create) architecture descriptions?

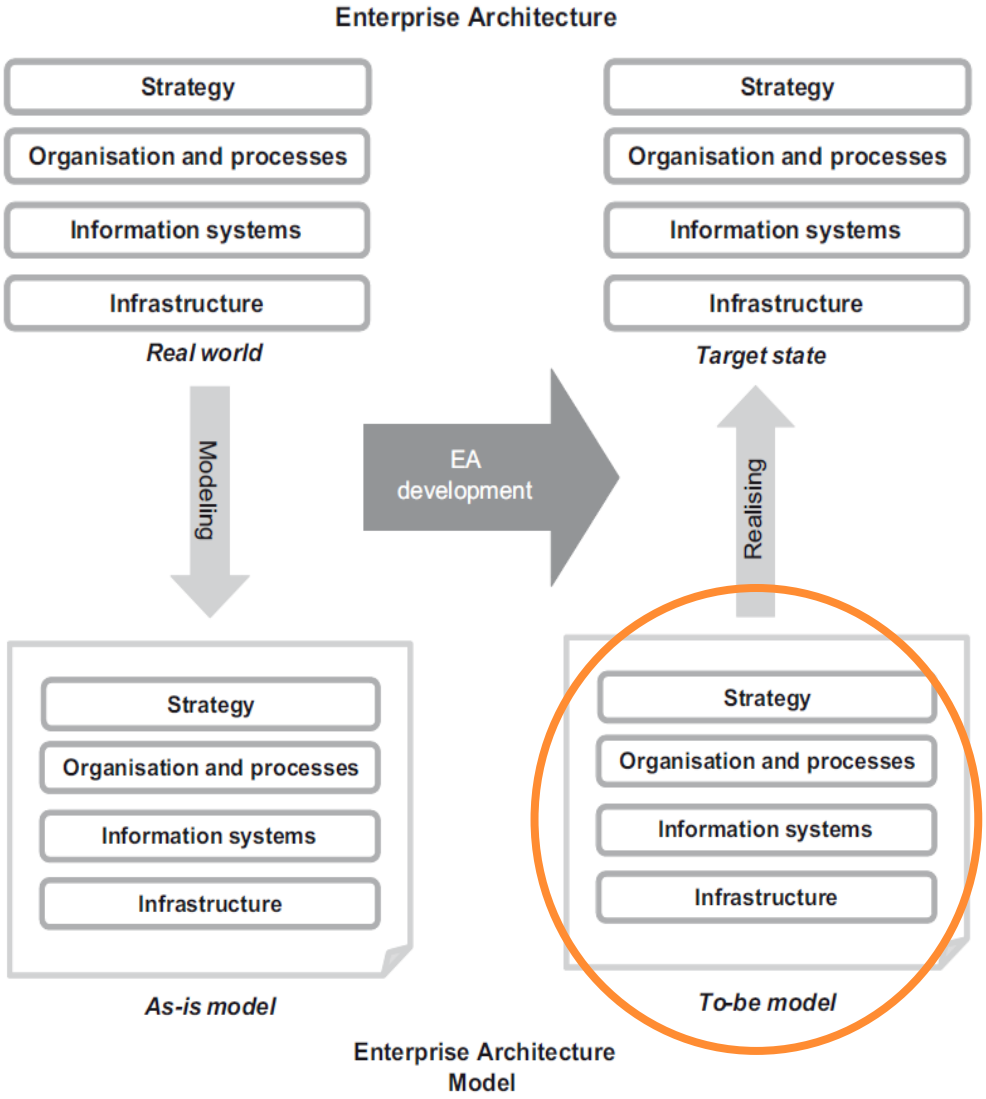
ⓘ Start presenting to display the poll results on this slide.

Architecture Description for Continuous Change

- If you don't retain the descriptive representations after you create them (or if you never created them in the first place) and you need to change the resultant implementation, you have only three options:
  - **Directly change the system** and see what happens. (High risk!)
  - **Recreate ("reverse engineer") the architecture** representations from the existing ("as is") implementation. (Typical for many projects - Takes time and costs money!)
  - Scrap the whole thing and **start over again**.
- Better:

  Retain description of your enterprise architecture

Change the Model before you Change the System!



Retain description of your enterprise architecture

Enterprise Architecture Repository