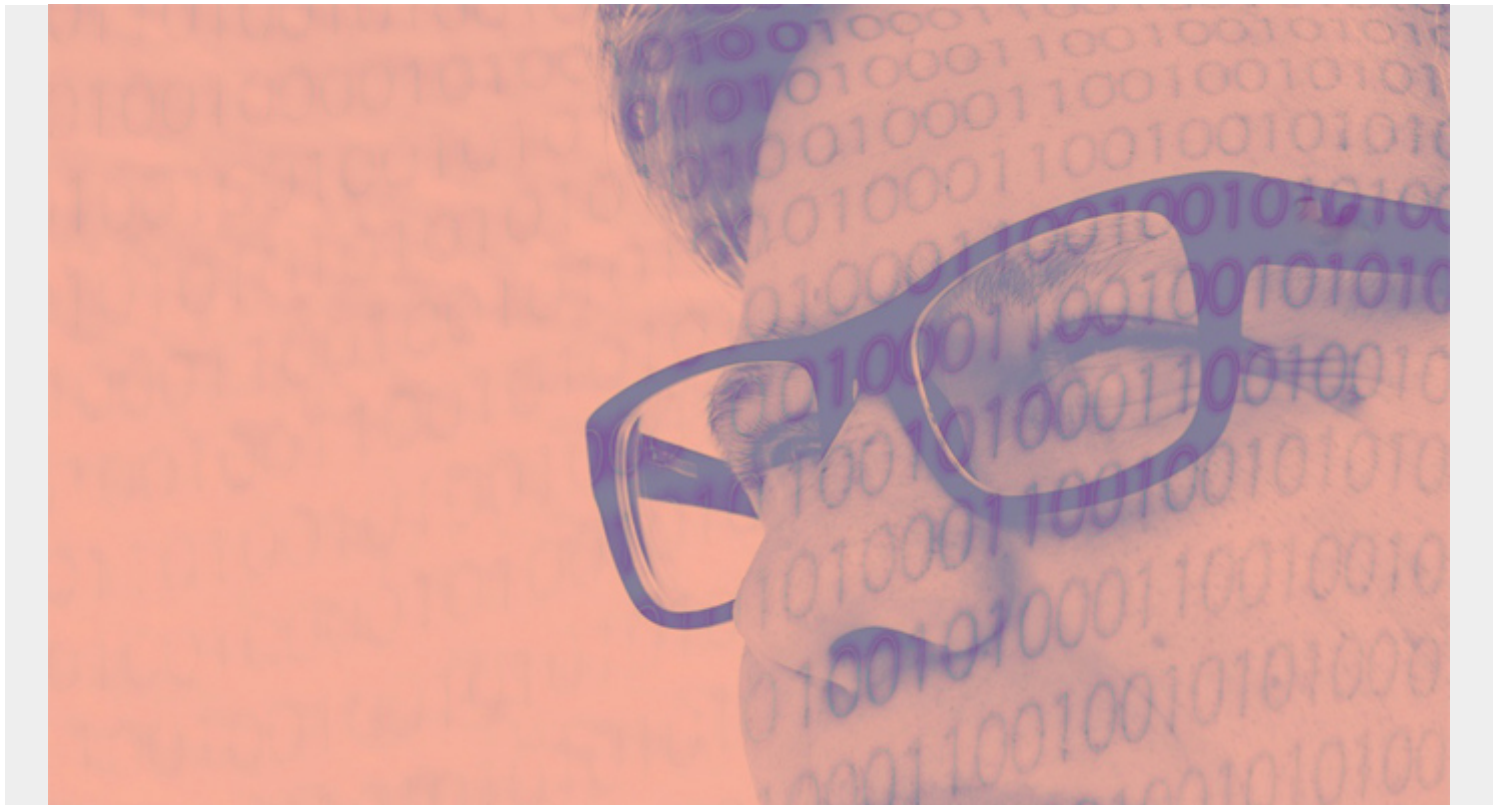# INTRODUCTION TO ZACHMAN FRAMEWORK

The Zachman framework was the brainchild of John Zachman in 1987, becoming a widely used approach for engineering Enterprise Architecture. It was published in the system journal of IBM under the name -- A framework for information systems architecture. Zachman worked for IBM from 1964-1990, serving as one of the founding developers of IBM's Business Systems Planning (BSP).

The basic idea behind the Zachman framework is that the same complex item can be described for different purposes in different ways, using different types of descriptions. The framework provides 36 necessary categories for completely describing anything, especially complex things like manufactured goods. The 36 categories are made up of six rows and six columns taking the form of a two-dimensional matrix.

The six rows of the framework are:

- Planner's View (Scope Contexts) - This view describes the business purpose and strategy, defining the playing field for the other views.
- Owner's View (Business Concepts) - This view reveals which parts of the enterprise can be automated.
- Designer's View (System Logic) - This view outlines how the system will satisfy the organization's information needs.
- Implementer's View (Technology Physics) - This is a representation of how the system will be implemented while addressing production constraints.
- Sub-Constructor's View (Component Assembles) - These representations illustrate the

implementation-specific details of certain system elements.
- User's View (Operations Classes) - This is a view of the functioning system in its operational environment.

The columns represent the interrogatives or questions that are asked of the enterprise.

- What (data) - What is the business data, information, or objects?
- How (function) - How does the business work by defining processes?
- Where (network) - Where are the business operations?
- When (time) - When are the business processes performed?
- Why (motivation) - Why is the solution chosen, how was it derived, and what motivates the performance of certain activities?

Rules of the Zachman Framework

Zachman defines 7 rules for using his framework.

- Rule 1: Do not add rows or columns to the framework.
  Thousands of years of linguistic experience would establish that Who, What, When, Where, Why, and How are the six primitive interrogatives. If you can answer all of these six questions, then you can derive answers to any other questions about the subject or object. Adding rows or columns to the framework would denormalize the classification scheme.
- Rule 2: Each column has a simple generic model.
  Each column of the framework is descriptive of a single, independent variable within the analytical target, in our case, the Enterprise. Therefore, the basic generic model of any one column is very simple: the variable (abstraction) it represents as related to itself.
- Rule 3: Each cell model specializes in its column's generic model.
  The specific model for any given cell will have to be customized to the constraints, the semantics, the vocabulary, the terms, and facts of the row's perspective. Furthermore, considering that the cell description forms the baseline for managing change, the (meta) model will have to express all of the concepts affected by changing to that cell model. Therefore, the specific (meta) model for a given cell will start with the generic, columnar model, be adjusted for the row's semantic constraints and then potentially be extended to accommodate all of the relevant concepts for expressing the constraints of the cell's row perspective as well as for managing change to the cell model itself.
- Rule 4: No meta concept can be classified into more than one cell.
  The framework constitutes a clean normalized classification system, with each column being unique. No meta concept can be classified into more than one cell. There is no redundancy. This is the one fundamental factor that makes the framework a good analytical tool.
- Rule 5: Do not create diagonal relationships between cells.
  First, the fact that the owners, designers, builders, and sub-contractors are all using the same word to mean entirely different things creates a very confusing communication problem. The people in the Enterprise may speak the same language and use the same words but may not be communicating effectively with each other. The structural reason for banning diagonals is because cellular relationships are transitive. Changing a cell logically may impact the cell above and below in the same column and every other cell in the same row.
- Rule 6: Do not change the names of the rows or columns.
  Do not change the names for the rows or columns, either in the generic framework or in the

Enterprise-specific framework. If you change the name of the rows and columns, you also change the meaning of the row or column affected. You can de-normalize the framework making it no longer comprehensive.

- Rule 7: The logic is generic and recursive.
  The logic of the framework is generic. It can be used to classify the descriptive representations of anything and to analyze anything relative to its architectural composition. It is recursive. It can be used to analyze the architectural composition itself. The framework is inert. It doesn't know what it is being used to analyze. Only the analyst knows the analytical target and establishes the boundaries of the analysis, and the analytical boundaries selected have far-reaching implications.

How and Where is the Zachman Framework Used?

In today's complex business environments, many large organizations have great difficulty responding to change. Part of this difficulty is due to a lack of internal understanding of the complex structure and components in different areas of the organization, where legacy information about the business is locked away in the minds of specific employees or business units.

The Zachman framework provides a means of classifying an organization's architecture. It is a proactive business tool, which can be used to model an organization's existing functions, elements, and processes while helping manage business change. The framework draws on Zachman's experience of how change is managed in complex products. John Zackman stresses that the framework can extend to the entire enterprise architecture, and is not just restricted to information architecture.

John Zachman sees the framework as being used as:

- A Planning Tool - helping you position issues and make informed choices.
- A Problem-Solving Tool - taming the complexity of the business abstraction, enabling the isolation of individual business variables.
- Evaluating tools and methods by mapping them to the framework and therefore proving a neutral way of cataloging what they do and do not support.
- A context for building up flexible, component architectures and systems capable of supporting high rates of enterprise change and replacing the "inventory of existing systems" which are "not integrated" as a result of being built "out-of-context."

Putting the Zachman Framework into Practice.

The logical point to begin within the Zachman framework would be at the top left of the two-dimensional matrix and work your way down across the table. The relevant business information or models used to represent a specific area of the business may already exist in business plans, project schedules, system specifications, procedure guidelines, or other documents.

As you go through the matrix, there will be gaps that need to be filled in where implicit information known to only a single person or a few experts needs to be made explicit and available to a wider audience. There may be instances of overlap or redundancy. The goal is to manage change and reduce redundancies and overlaps.