



Software Architecture Frameworks

Software Architecture
3rd Year – Semester 1
Lecture 16

Why Software Architecture

- Provides standard governing structure
- Provide solutions to known problems
- Helps to make projects successful
 - Addresses: Failing to consider key scenarios, failing to design for common problems, or failing to appreciate the long term consequences of key decisions can put your application at risk
- Makes products easy to maintain

When do we need to “Architect”

- When the Solution(s) gets bigger
 - Modern Software are more complex than what was before
- When you have to think about the future
 - Software lasts longer, they are no longer “throw-away” items - specially on Data
- Increased Usage and Usage Types
 - Earlier only direct users interacted with Software, now Systems interact with each other

Enterprise Architecture

- A well-defined practice for conducting enterprise analysis, design, planning, and implementation, using a holistic approach at all times, for the successful development and execution of strategy
- Applies architecture principles and practices to guide organizations through the business, information, process, and technology changes necessary to execute their strategies

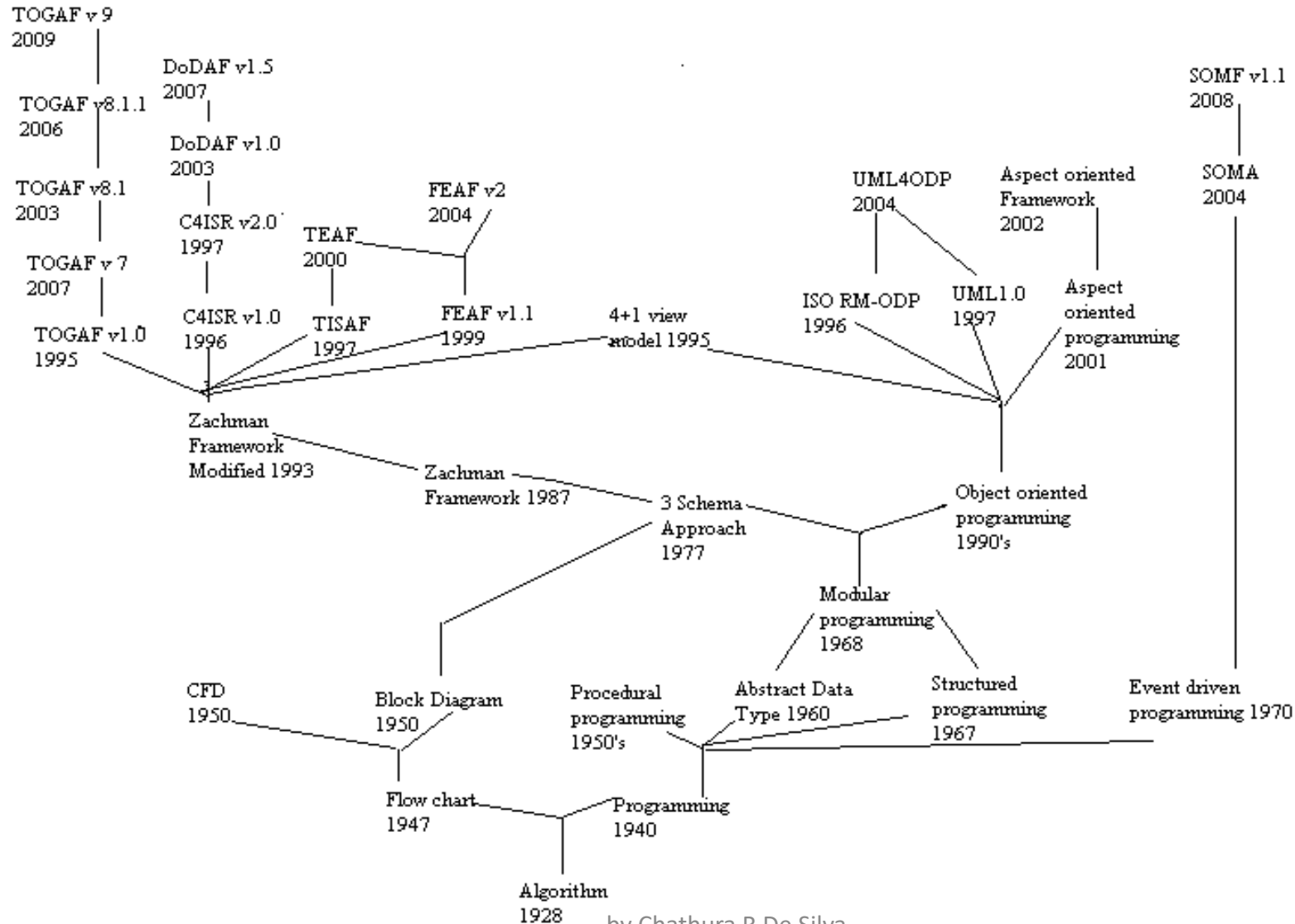
Benefits of Enterprise Architecture

- Business Benefits
 - Helps an Organization achieve its business strategy
 - Faster time to market for new innovations and capabilities
 - More consistent business process and information across business units
 - More reliability and security, less risk
- IT Benefits
 - Better traceability of IT costs
 - Lower IT costs – design, buy, operate, support, change
 - Faster design and development
 - Less complexity
 - Less IT risk

Key Architecture Principles

- Build to change instead of building to last
 - Consider how the application may need to change over time to address new requirements and challenges
 - Build with flexibility to adopt changes
- Model to analyze and reduce risk
 - Use design tools to visualize e.g. UML
 - Capture requirements and architectural and design decisions and to analyze their impact
 - Do not formalize the model to the extent that it suppresses the capability adapt easily
- Communication and Collaboration
 - Use visualizations of the architecture to communicate and share your design efficiently with all the stakeholders, and to enable rapid communication of changes to the design
- Identify key engineering decisions
 - Understand the key engineering decisions and the areas where mistakes are most often made
 - Invest in getting these key decisions right the first time (late changes are always costly)

Architecture Frameworks - How it evolved



Enterprise Architecture Frameworks

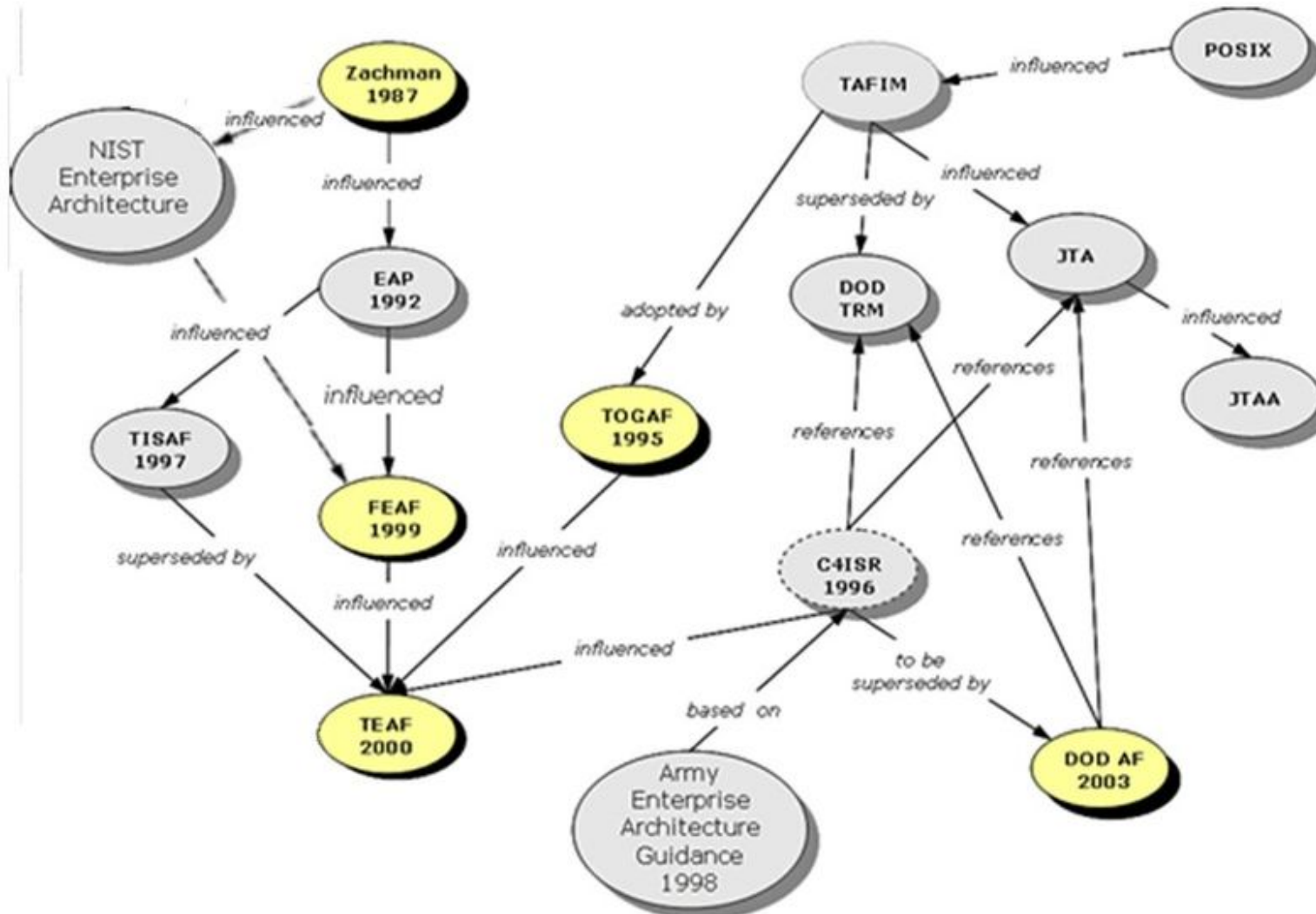
- Zachman Framework

- In 1982, when working for IBM and with BSP, John Zachman was perhaps the first to mention Enterprise Architecture in the public domain. In 1987, John Zachman, who was a marketing specialist at IBM, published the paper, A Framework for Information Systems Architecture. The paper provided a classification scheme for artifacts that describe the what, how, where, who, when and why of information systems.

- The Open Group Architecture Framework (TOGAF)

- In 1994, the Open Group selected TAFIM from the US DoD as a basis for development of The Open Group Architecture Framework (TOGAF), where architecture meant IT architecture.

Architecture Frameworks – Mutual Influence

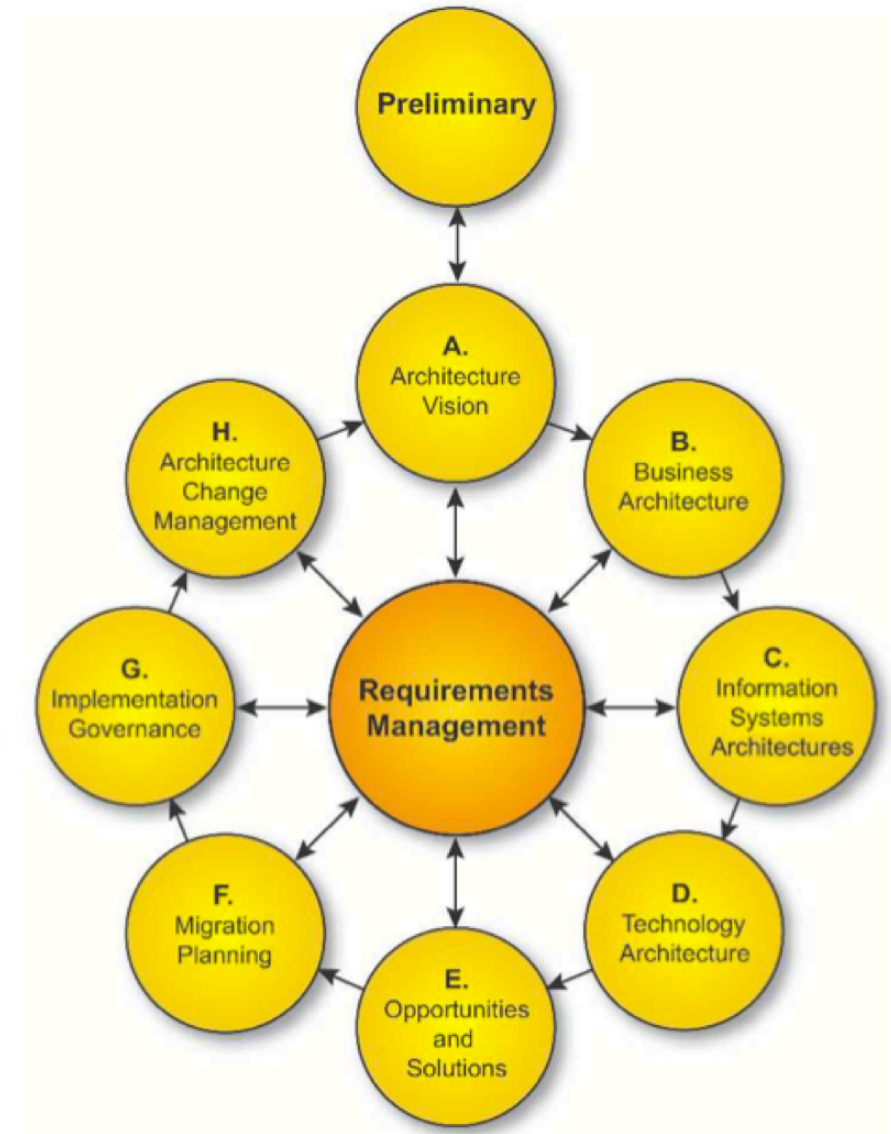


TOGAF

- The Open Group Architecture Framework (TOGAF) is a framework for enterprise architecture that provides an approach for designing, planning, implementing, and governing an enterprise information technology architecture.
- TOGAF is a high level approach to design. It is typically modeled at four levels: Business, Application, Data, and Technology.
- TOGAF Relies heavily on modularization, standardization, and already existing proven technologies and products
- TOGAF Components
 - Architecture Development Method (ADM)
 - Enterprise Continuum
 - Resource Base

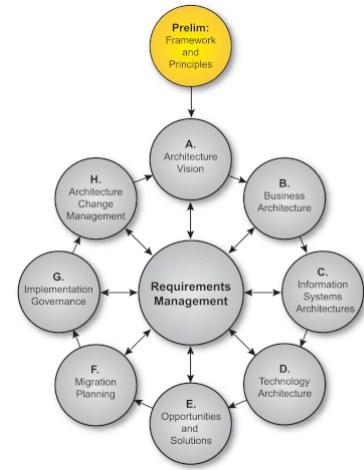
TOGAF – Architecture Development Method (ADM)

- A Step by Step process to Developing or Changing an Architecture
 - Preliminary } Defines the need for Architectural Change
 - Phase A }
 - Phase B } Develops a Clear description of the Future Architecture
 - Phase C }
 - Phase D }
 - Phase E } How the Vision & Future Architecture is Delivered
 - Phase F } Implementation & Migrations Planning
 - Phase G } Architectural Oversight to Implementation
 - Phase H } Architecture Change Management
 - Requirements Management

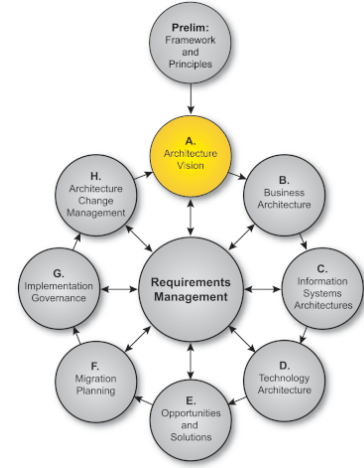


ADM – Preliminary Phase

- Defines what needs to be done, how it will be carried out.
- Establish parameters for a successful iteration of ADM
- Identify and establish architecture Frameworks & Principles
- TOGAF can be tailored to meet the needs
- TOGAF can be integrated with other management frameworks (e.g. PRINCE2)
- Output: Request for Architecture Work
 - Outlines Requirements, Organizational Context, Structures, Tools or Architecture Frameworks

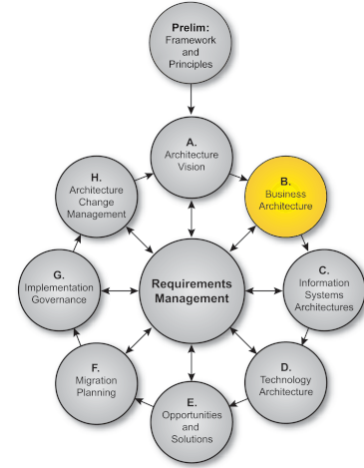


ADM – Phase A: Architecture Vision



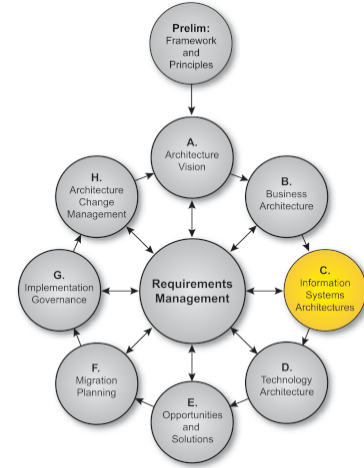
- Starts with Request for Architecture Work
- Sells the benefits of the proposed capability to stakeholders and decision-makers
- Outlines Vision for the Architecture
 - High-Level aspiration of capabilities
 - Business values that the Architecture will deliver
- Identifies Concerns and Requirements.
- Confirms business goals, drivers and constraints
- Goal is to make sure that the enterprise is Able, Ready, Willing and Committed to make the necessary Architecture Changes
- Output: Statement of Architecture Work
 - It also provides the Vision of the proposed enterprise architecture. This sense of direction is vital for guiding the work throughout this iteration of the ADM

ADM – Phase B: Business Architecture



- Starts with Improving Business Capability
- Key objective is to identify Target Business Architecture that shows how the enterprise can achieve the Architecture Vision
- Business Architecture demonstrate Business Value and Return on Investment (ROI) to the Stakeholders
- Output:
 - Business models
 - Activity or Process models
 - Use Case

ADM – Phase C: Information Systems Architecture



- Takes the Business Perspective from previous Phase as the input
- Information Systems Architecture compose Data Architecture and Applications Architecture
- Data & Application Architecture uses different reference models
 - Data Architecture – Class Diagrams, ER Diagrams
 - Application Architecture – Application Communication, Component Diagram, etc...
- Identify Candidate Architecture Roadmap Components
- Output:
 - Architecture Definition Document

Exercise #1

- Business Case:
 - There are many types of Vehicles; Cars, Vans, Bikes, etc... Different types of vehicles may have specialized attributes but there are a few attributes in common.
 - A Person may own a vehicle for a given time and this ownership is registered at the Department of Motor Vehicles.
- Q1) How would you develop Data Architecture for the Above? State your assumptions.
- Q2) Draw the Data Architecture diagrams.

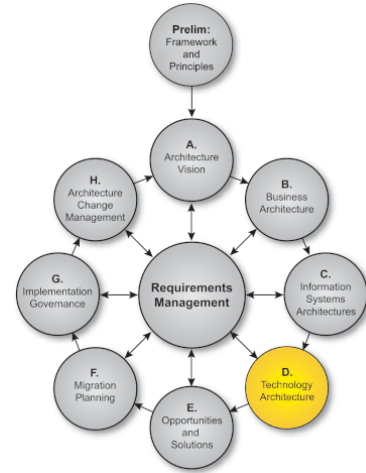
Exercise #1: Sample Answers...

- Developing Data Architecture
 - Requirement Clarification meetings to be held to better understand the requirement. Use Case Diagrams to be created
 - A Gap Analysis on the requirement & business entities needs to be created.
 - Business and Logical Data Models to be identified. Create Diagrams i.e. Class Diagrams / Entity Relationship Diagrams to be created.
- Assumptions
 - A vehicle cannot be a Car or a Van at the same time [Total Participation]
 - Vehicle will be owned by only 1 person at a given time [Person : Vehicle = 1 : M]
 - A vehicle ownership can be transferred from one person to another [Association Vs. Composition]
 - Vehicle & Person attributes can be represented in Text, Number and Date formats
- Diagrams
 - ER Diagram
 - Class Diagram
 - Table Structures

Exercise #2:

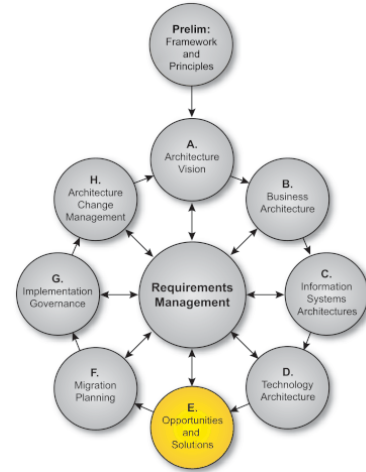
- Business Case:
 - A CCTV Camera system captures a set of videos and send them to a central Server. The server needs to store the video and meta information for later retrieval.
- Q1) Create the Data Architecture
 - Data Architecture diagrams.
- Q2) Create the Application Architecture
 - Application Communication
 - Component Diagram

ADM – Phase D: Technology Architecture



- Technology Architecture is a description of...
 - Structure and interaction of the platform services
 - Logical and Physical technology components.
- Develop Baseline Technology Architecture
 - Creates Technology Reference Models & Criteria for Measurement
 - Develop Target Technology Architecture - requirements traceability, criteria for selection of service portfolio
- Output:
 - Baseline Technology Architecture
 - Networked Computing/Hardware view
 - Communications view
 - Processing view
 - Technology Architecture Report (summarizing the key findings)

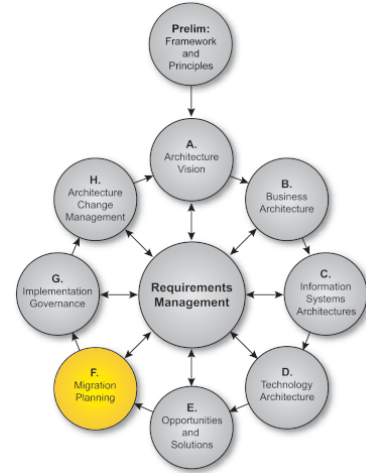
ADM – Phase E: Opportunities and Solutions



- About finding Opportunities for...
 - Delivering the Target Architecture by implementing specific Solutions.
 - Concentrated on How to Deliver the Architecture
 - When the change is large, this Phase provides an Incremental Approach to convert from Baseline to Target Architecture
- Generates the first complete version of Architecture Roadmap by combining the analysis and suggestions from the Architecture Development phases
- Output:
 - High-level Implementation Plan
 - High-level Migration Plan & Impact Analysis

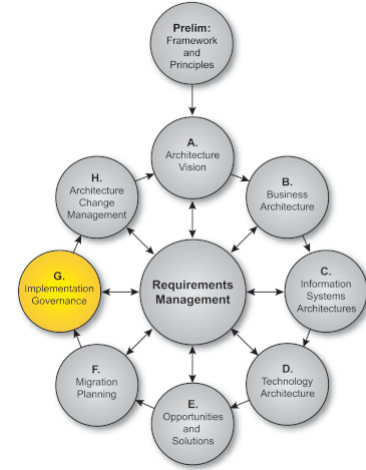
ADM – Phase F: Migration Plan

- Finalizes a detailed Implementation and Migration Plan
 - Also finalizes the Architecture Roadmap
- Plan is coordinated with...
 - Change management approach used within the enterprise
 - Business Planning
 - Enterprise Architecture
 - Portfolio and Project Management
 - Operations Management
- Goal is to ensure that key stakeholders fully understand
 - Business value
 - Cost of work packages
 - Transition and Future Architectures



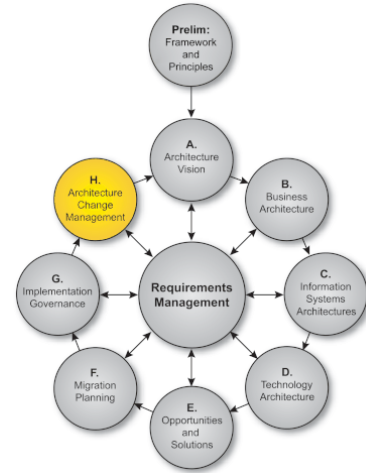
ADM – Phase G: Implementation Governance

- Provides an Architectural Oversight to Implementation
 - Ensures Project Implementation conforms the Target Architecture
- Formulate Project Recommendations
- Manages Implementation Driven Architecture Changes
- Review Ongoing Implementation Governance and Architecture Compliance
 - Confirms the scope and priorities for deployment
 - Guiding development and solutions deployment
 - Performs compliance reviews
- Output:
 - Architecture Contract Document (drives any Architecture Changes)

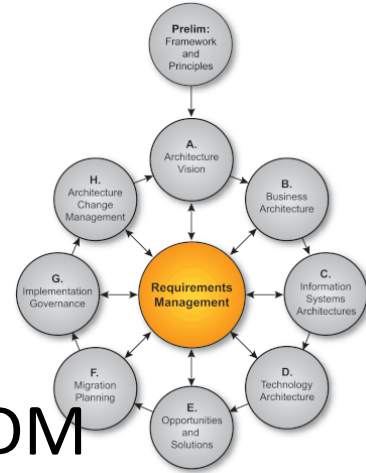


ADM – Phase H: Architecture Change Management

- Change management process to manage changes to the Architecture
 - Process for managing changes
 - Ensure Architecture achieves its intended Business Value
- Requires continues monitoring
 - Governance Requests
 - New Technologies
 - Changes in the Business Environment
 - Strategic Changes (e.g. Cost)
- Judge whether a Change Request warrants a simple architecture update or whether it requires to Re-Architect with ADM
- Output:
 - Architecture updates
 - Changes to architecture framework and principles
 - New Request for Architecture Work



ADM – Requirement Management

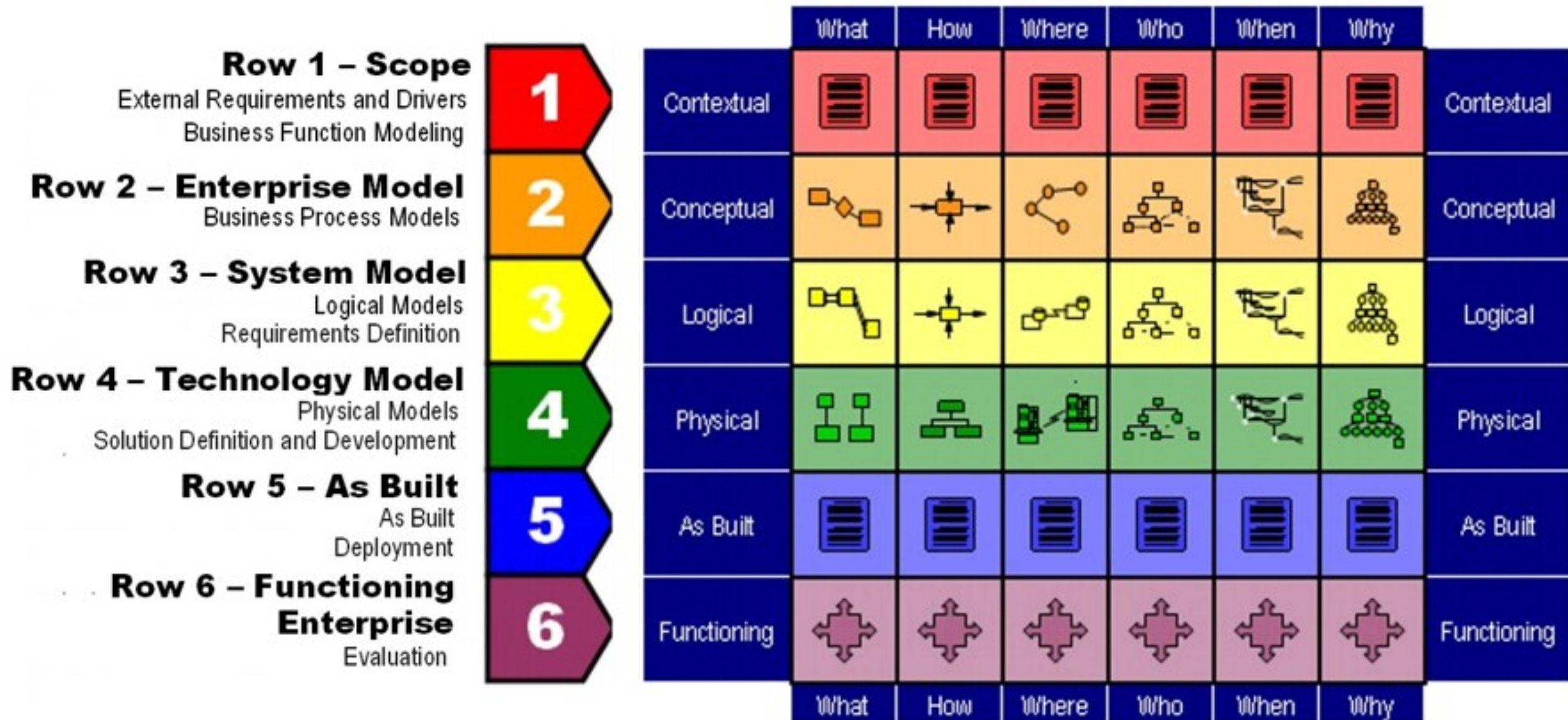


- It's a Continuing Ongoing Process and sits in the center of the ADM
 - Requirements are Produced, Analyzed and Reviewed in each ADM Phase
- Ensured Changes to Requirement are well governed and Reflected in all other Phases
- Describes a Process for Requirements Management and how they are lined to the other Phases
- Output:
 - Changed requirements
 - Requirements Impact Statement

Zachman Framework

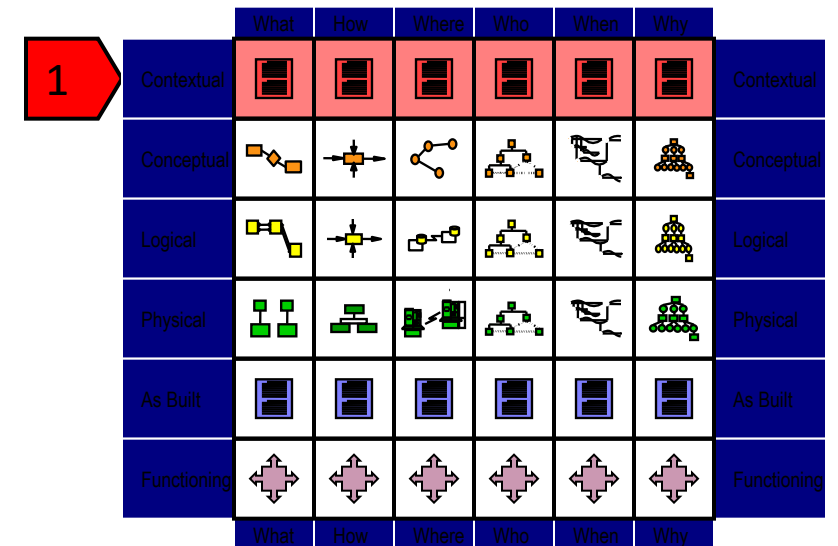
- Zachman Framework is an enterprise ontology and is a fundamental structure for Enterprise Architecture which provides a formal and structured way of viewing and defining an enterprise.
- The ontology is a two dimensional classification schema that reflects the intersection between two historical classifications.
 - Dimension #1: What, How, Where, Who, When and Why
 - Dimension #2: Contextual, Conceptual, Logical, Physical, As Built and Functioning Enterprise

Zachman Framework – Rows



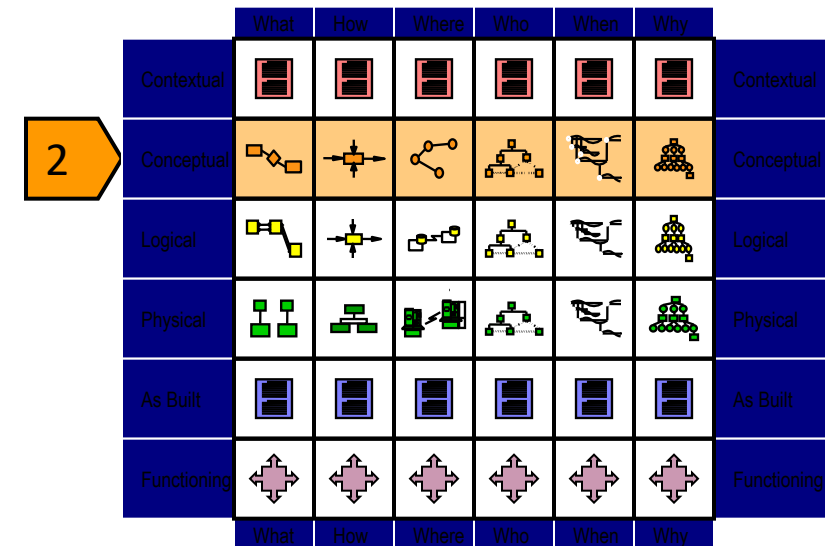
Zachman – Row 1: Scope (Planner's View)

- Motivation/Why
 - Business goals, objectives and performance measures
- Function/How
 - High-level business functions
- Data/What
 - High-level data classes related to each function
- People/Who
 - Stakeholders related to each function
- Network/Where
 - Locations related to each function
- Time/When
 - Cycles and events related to each function



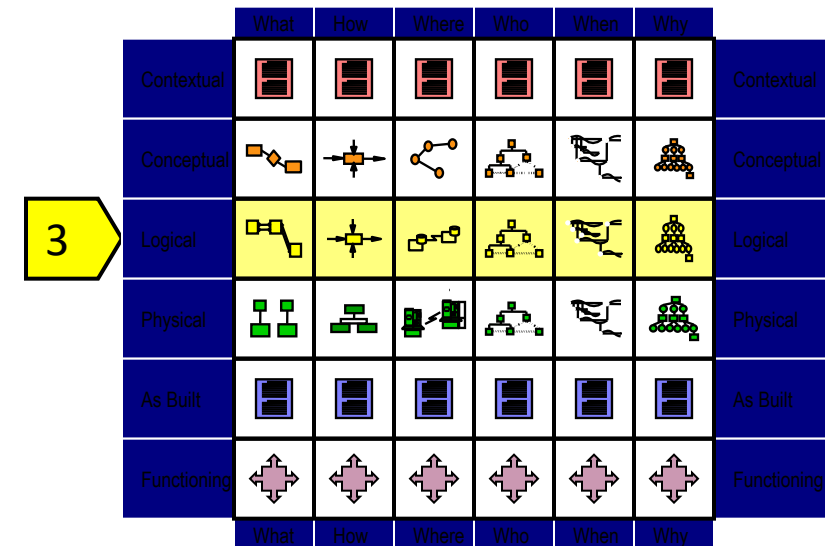
Zachman – Row 2: Enterprise Model (Designer's View)

- Motivation/Why
 - Policies, procedures and standards for each process
- Function/How
 - Business processes
- Data/What
 - Business data
- People/Who
 - Roles and responsibilities in each process
- Network/Where
 - Locations related to each process
- Time/When
 - Events for each process
 - Sequencing of integration
 - Process improvements



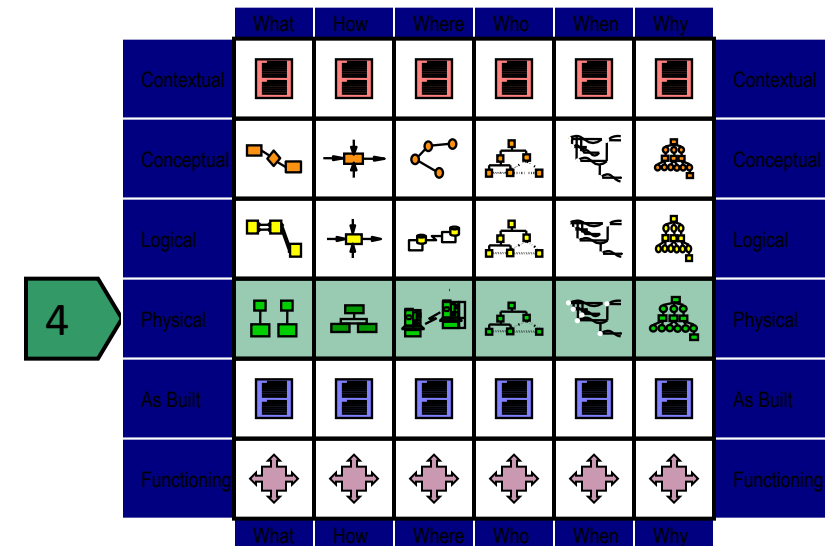
Zachman – Row 3: System Model (Designer's View)

- Motivation/Why
 - Policies, standards and procedures associated with a business rule model
- Function/How
 - Logical representation of information systems and their relationships
- Data/What
 - Logical data models of data
 - Data relationships
- People/Who
 - Logical representation of access privileges
- Network/Where
 - Logical representation of the distributed architecture
- Time/When
 - Logical events and their triggered responses



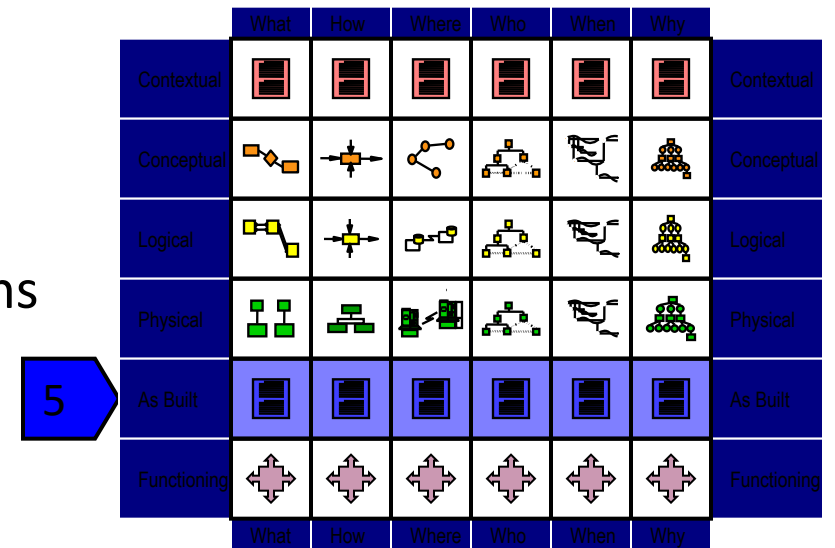
Zachman – Row 4: Technology Model (Builder's View)

- Motivation/Why
 - Business rules constrained by information systems standards
- Function/How
 - Specifications of applications that operate on particular technology platforms
- Data/What
 - Database management system
 - Logical data models
- People/Who
 - Access privileges to technologies
- Network/Where
 - Network devices and their relationships
- Time/When
 - Specification of triggers to respond to system



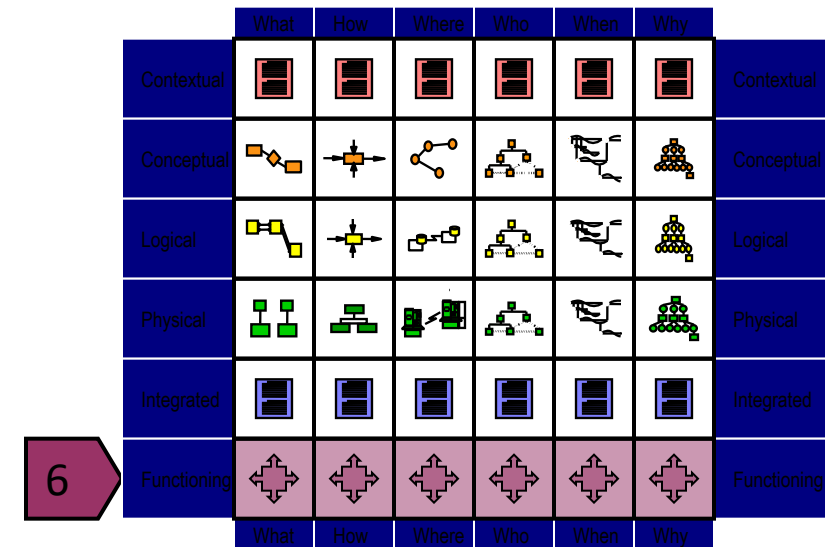
Zachman – Row 5: As Built (Integrator's View)

- Motivation/Why
 - Business rules constrained by specific technology standards
- Function/How
 - Programs coded to operate on specific technology platforms
- Data/What
 - Data definitions by physical data models
- People/Who
 - Access privileges to control access
- Network/Where
 - Devices configured to conform to node specifications
- Time/When
 - Timing definitions coded to sequence activities



Zachman – Row 6: Functioning Enterprise (User's View)

- Motivation/Why
 - Operating characteristics of specific technologies constrained by standards
- Function/How
 - Functioning computer instructions
- Data/What
 - Data values stored in actual databases
- People/Who
 - Personnel and key stakeholders / roles
- Network/Where
 - Sending and receiving messages
- Time/When
 - Timing definitions operating to sequence activities



Zachman Framework – Outputs

| ENTERPRISE | What | Where | Why | How | Who | When |
|------------|---------------------|-------------------|-------------------|----------------------|----------------------|--------------------|
| Conceptual | Entity Relations | Node Relations | Goal Relations | Process Relations | Persona Relations | Event Relations |
| Contextual | Entity Associations | Node Associations | Goal Associations | Process Associations | Persona Associations | Event Associations |
| Logical | Entity Attributes | Node Attributes | Goal Attributes | Process Attributes | Persona Attributes | Event Attributes |
| Physical | Entity Domains | Node Domains | Goal Domains | Process Domains | Persona Domains | Event Domains |
| Mechanical | Entity Definitions | Node Definitions | Goal Definitions | Process Definitions | Persona Definitions | Event Definitions |
| Instantial | Entities | Nodes | Goals | Processes | Personas | Events |

References

- <http://pubs.opengroup.org/architecture/togaf8-doc/arch/toc.html>
- <https://www.orbussoftware.com/enterprise-architecture/togaf/what-is-the-adm/>
- <https://www.zachman.com/>