System(s) Architecture Practices

Everyone claims:
to do it,
has done it,
or knows how

Architecture Definition

"An architecture is the fundamental organization of a system embodied in its components, their relationships to each other, to the environment, and the principles guiding its design and evolution."
IEEE STD 1471-2000

What is an "Architecture"?

An architecture consists of:

- Requirements that are partitioned into functions that are mapped or assigned– Objective System(s) → System Family(ies)/SW → Sub-systems → Components → Sub-Components → Artifacts
- The relationships among the elements – interfaces, links, rules for interaction, and interdependencies
  - "What will interact with what; how; why" ?
  - Guidelines governing their design and evolution over time.
- Architecture is the more general abstraction of; objective functions, expected capabilities, and processes around which design is centered
- Architecture is the first deliverable of the software development life cycle
- An architecture facilitates "system solutions and system design" at multiple levels of abstraction and states.
- An architecture is comprised of one or more "views" and is documented as such

What Comprises an "Architecture"?

An architecture consists of:

- Requirements that are partitioned into functions that are mapped or assigned to the intended operations space and decomposed.
- The relationships among the elements – interfaces, links, rules for interaction, artifacts, interdependencies, etc.
  - "What will interact with what; how; why" ?
  - Guidelines governing their design and evolution over time.
- The more general abstraction of;
  - objective functions,
  - expected capabilities, and
  - processes

  around which design is centered.

An Architecture Representation Is Important to Both Design and Development

- First deliverable of the software development life cycle
- Facilitates "system solutions and system design"
  - at multiple levels of abstraction and states, interchangeably and simultaneously.
- Expressed as one or more "views" and is documented as such

Different Frameworks for Different Purposes

The following subset of the most common architecture frameworks was considered:

- **Zachman Institute for Framework Advancement (ZIFA)**
- **Federal Enterprise Architecture Framework (FEAF)**
- **Meta Group Architecture Framework (MGAF)**
- **The Open Group Architecture Framework (TOGAF)**
- **Treasury Enterprise Architecture Framework (TEAF)**
- **Department of Defense Architecture Framework (DoDAF)**

Software System Architecture Products

- **Graphical, textual, and tabular items that are developed in the course of understanding the precise performance requirements and functional capabilities of building a given software capability.**
- **All architecture products are empirically grounded.**

Why Architecture Analysis?

- **Achieve architectural integrity**
- **Achieve referential integrity**
- **Support requirements integrity**
- **Provides documentation of the system(s) SW designs**
- **Support understanding and management of the Objective System(s) technical and operational requirements, and complexity.**
- **Helps maintain real-time, life cycle configuration management**
  - **It is required**
- **Supports the scale and virtualized understandings of the context within which specific models lay.**

Federal Policy and Guidance on Architecture Mandate for Architectures

Executive Order 13011: Federal Information Technology (1996).
Executive Agencies shall appoint a Chief Information Officer and "shall refocus information technology management to support directly their strategic missions, implement an investment review process that drives budget formulation and execution for information systems, and rethink and restructure the way they perform their functions before investing in information technology to support that work."

Information Technology Management Reform Act (ITMRA)/Clinger-Cohen Act. (1996)
Mandates that Chief Information Officers of Executive Agencies are responsible for "developing, maintaining, and facilitating the implementation of a sound and integrated information technology architecture for the executive agency ".

Chief Information Officer
Review and provide recommendations to the organization leadership organization budget requests for information technology and national security systems.

Policy Enabled by Common Approach for Developing Architectures

The Open Group Architecture Framework (TOGAF)

Common approach to develop enterprise architecture

Compliant Architecture Framework

Common approach for developing a system architecture description

Core Architecture Data Model (CADM)

Common data model for architecture data

Architecture Repository System (ARS)

Common repository for storing and retrieving architecture data

DoD Framework
One Architecture ... Three Views + 1

Operational View

Describes and interrelates the operational elements, tasks and activities, and information flows.

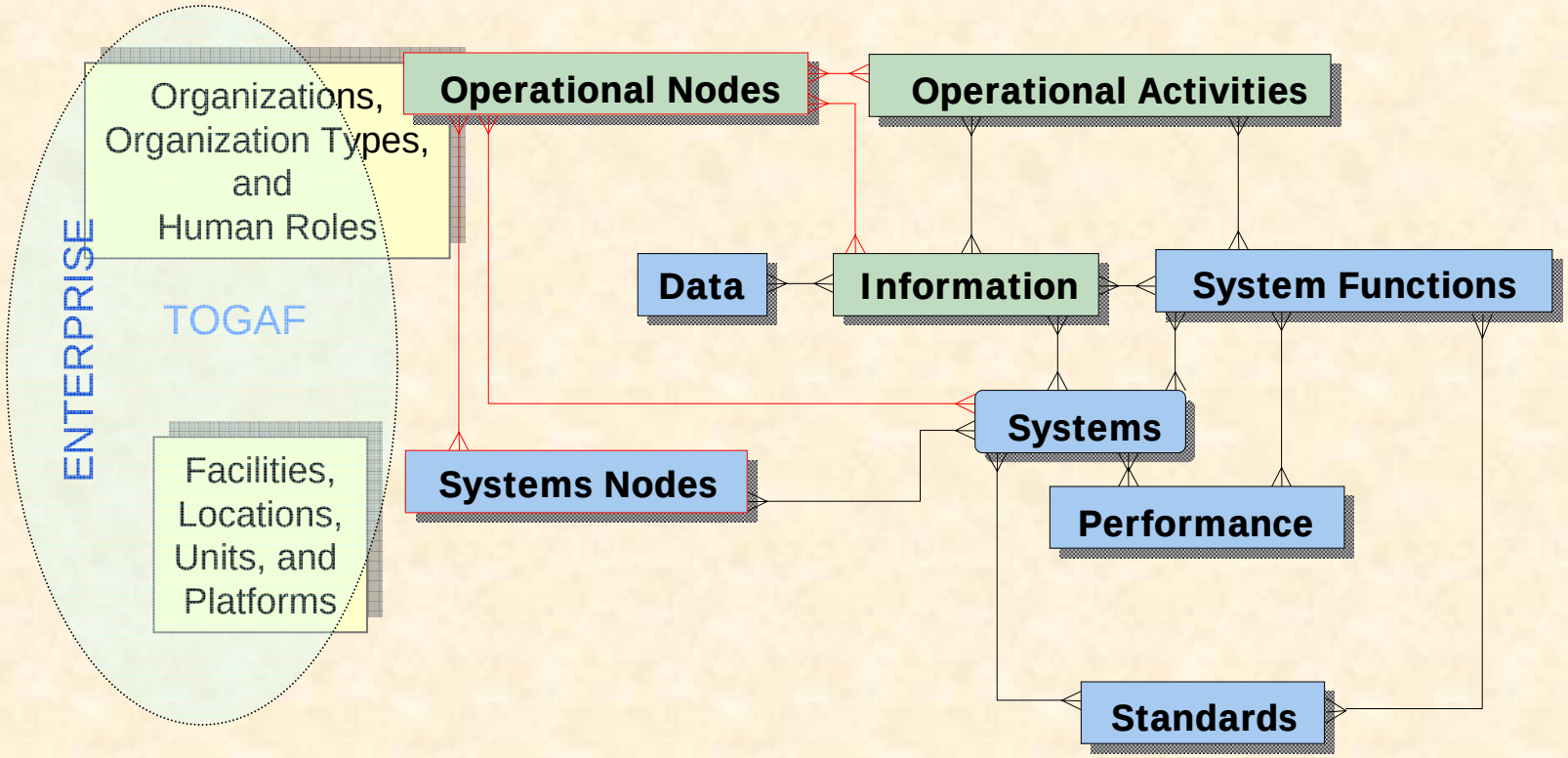
Systems View

Describes and associates systems, their interconnections and performance to the operational view and its requirements.

Technical View

Describes the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements.

Architecture Framework Core Elements



AF Product Elements, CADM Model, & Basis for ARS Schema

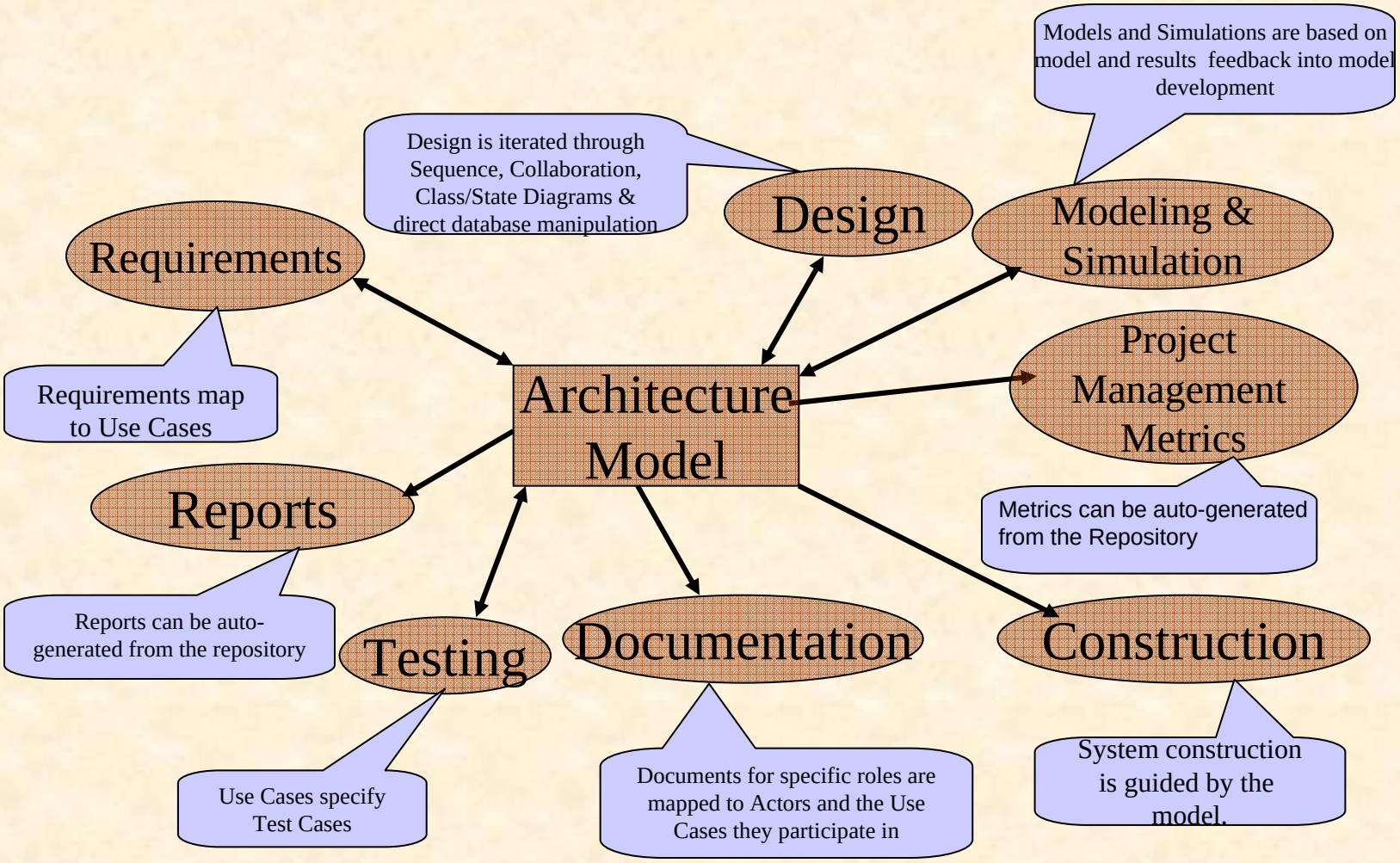
Some Benefits of Architecture Standards and Principles

- **Manage system complexity**
- **Manage system evolution**
- **Represent and optimize the system and requirements from each constituent point of view**
- **Manage requirements and trade-offs among systems, segments and subsystems, etc**
- **Align investment decisions within program budget**
- **Control requirements creep**
- **Evolutionary construction of computational engine through automation**
- **Facilitate life cycle changes/upgrades**

Some Architecture Stakeholders

- **Designers**
- **Developers**
- **Audit Agencies**
- **Cross-domain programs**
- **Sponsor(s)**
- **FoS Product IPTs**
- **IS&T IPT**
- **SE – Functional and SoS Analyses**
- **SE – Requirements**
- **SE Distributed Systems Capabilities**
- **T&E**
- **Complementary Systems**
- **SE – Architectural & Integrity Analyses**

Model Driven Development



Architecture Description Document

Structure

ADD
Common View Products
OVs
SVs

Appendix A
System(s) Software Architecture

Annex A-1 *IP 1*

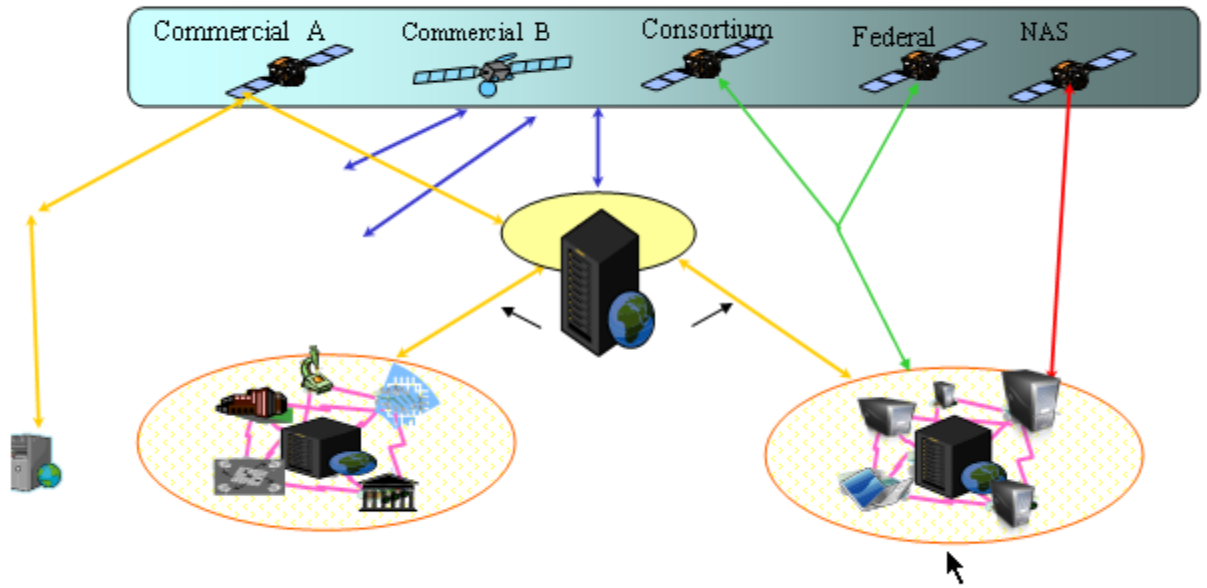
Annex A-2 *IP 2*

Annex A-3 *IP 3*

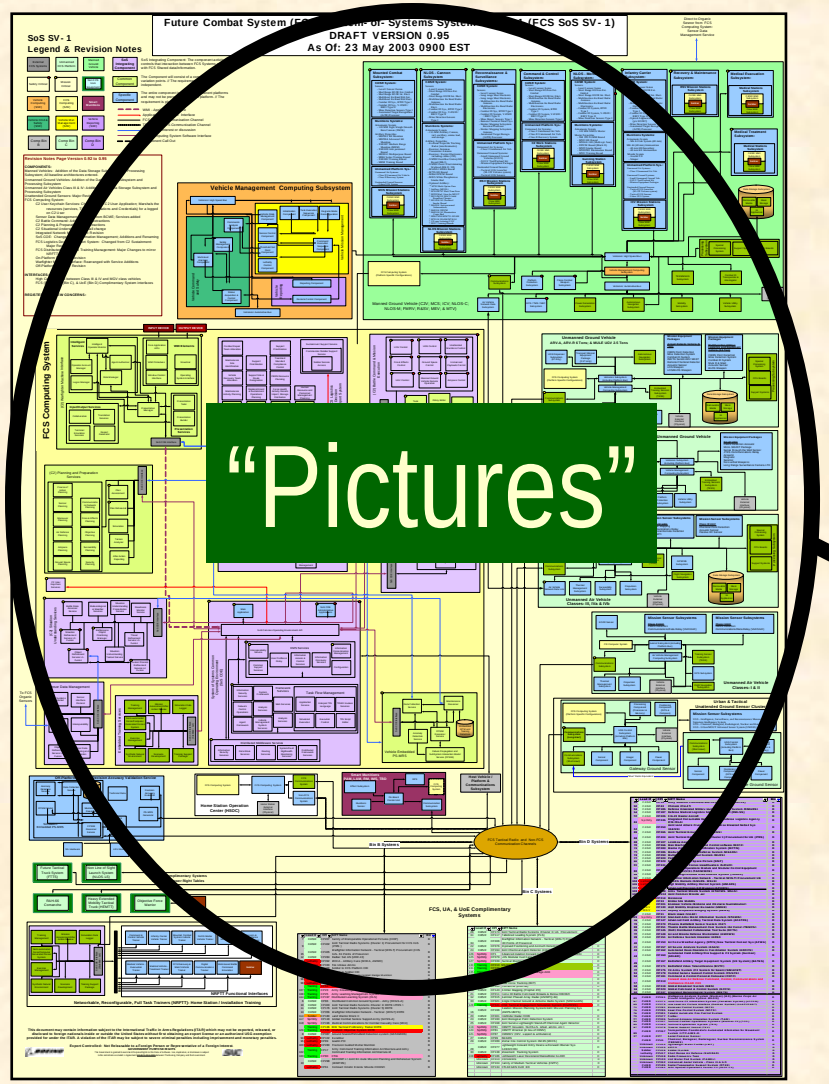
Annex A-4 *IP4….*

Annex 1-*N*
Integrated Process Products
OVs
SVs

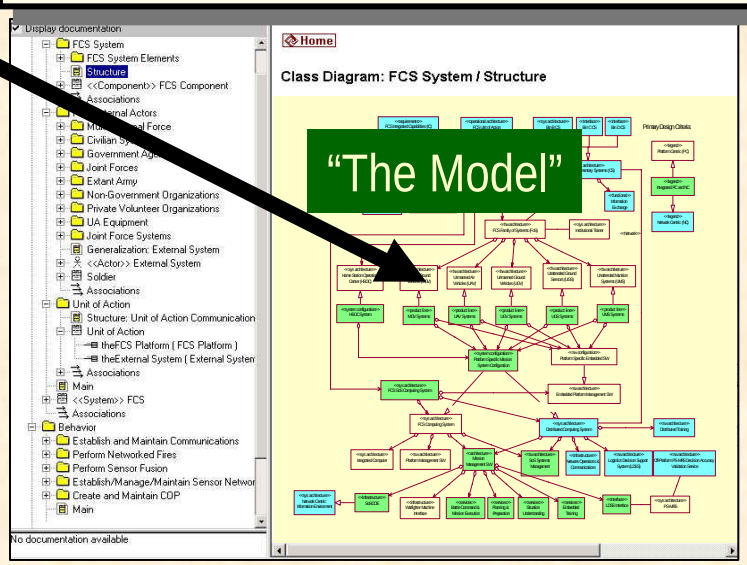
Inter Interface Architecture



Engineered System(s): Architectures



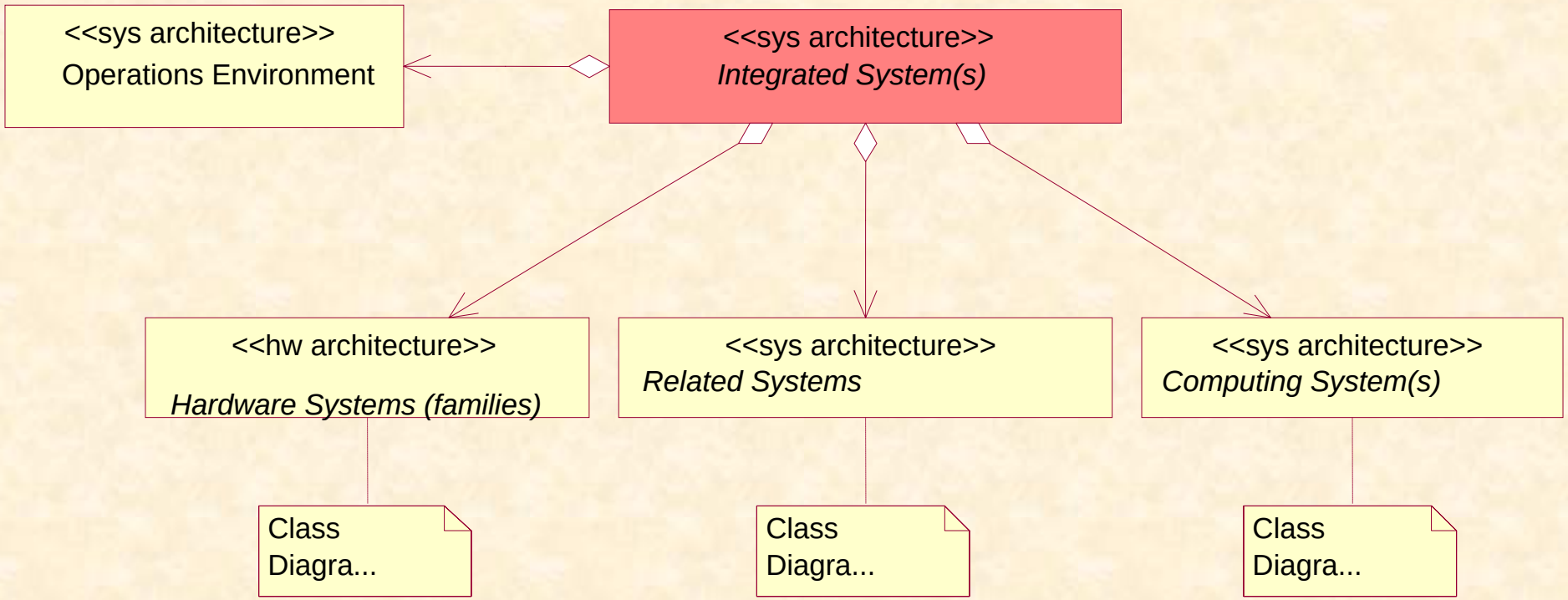
- **Detailed Architecture**
- **Developed in concert with Customer and Stakeholders**
- **Provides Logical and Physical infrastructure to support Integrated Process(es) – (Capabilities)**
- **Capture using modern Engineering & Integration Tools**
- **One Architecture : One Model – Many Development Views**



Architecture Model (Formal Methods Based)

- Shows how the system works
- Finds holes in requirements
- Finds duplication
- Provides an analysis framework
  - Functional analysis
  - Performance analysis
- Manages complexity
- Allows multiple integrated views of the system
- Creates integrated database of all (sub-)systems
- Provides a structured design methodology
- Identifies test cases
- Provides user and system documentation
- Specifies interfaces
- Model Driven Development

System(s) Architecture
(Integrated)



Take Away: System Architecture Purpose

- **Define and model high-level structure,** evolve to executable code.
- **Build objective system in a way that will incorporate cross component & sub-system integration as the primary tenet of its architecture.**
- **Balance cost and model expectations of all and each stakeholder (capabilities) that:**
  - **meet operational requirements**
  - **provide the entire constituency with flexible and adaptable means to close with and carry out assigned missions.**
  - **assure integration such that common and unique artifacts are defined once and employed ubiquitously**
  - **meet decomposed functional requirements**

Take Away: System(s) Architecture Description

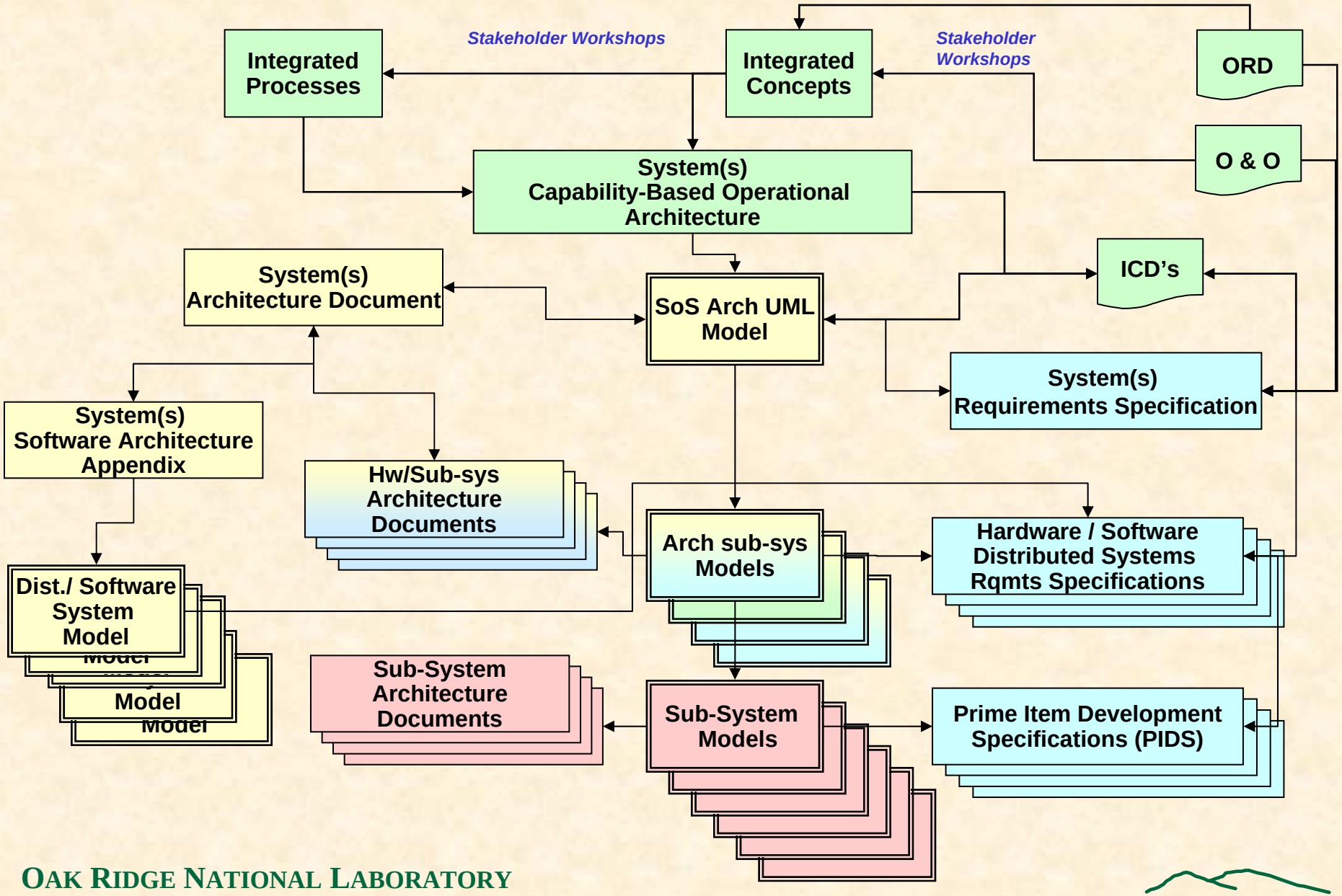
The system(s) architecture represents the hardware and software their, respective, component parts

- **What each do**
- **How each part relates to each other part**
- **The rules and constraints under which each part and the system of parts function**

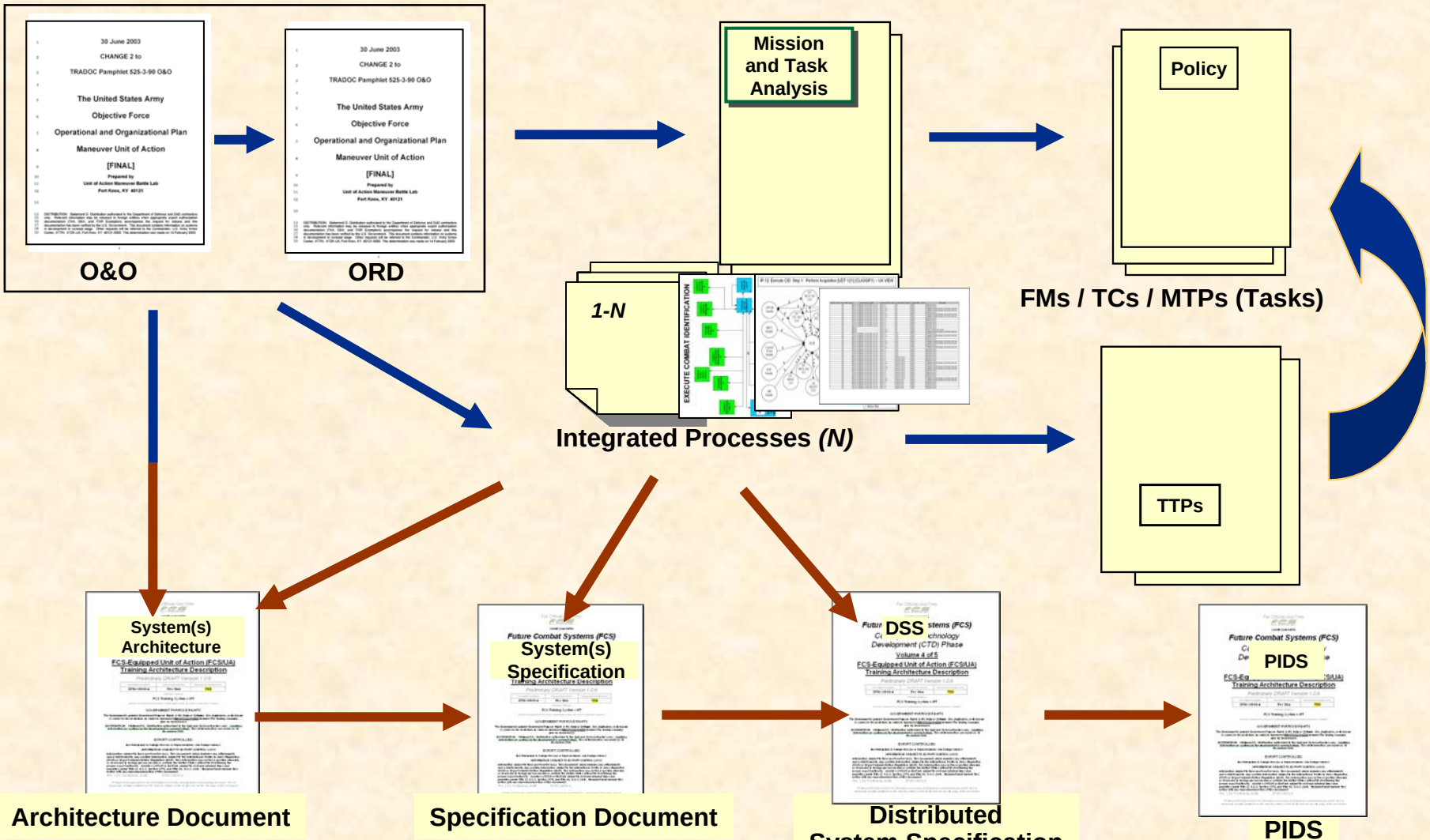
Included within the architecture is a formal description of the integration with other systems which may be important to objective system performance but outside its scope/domain. Examples might include:

- **other agency systems and results**
- **variously classified data/elements**
- **data and functionality incompatible with objective system(s)**
- **global distributive properties secondary to primary user**

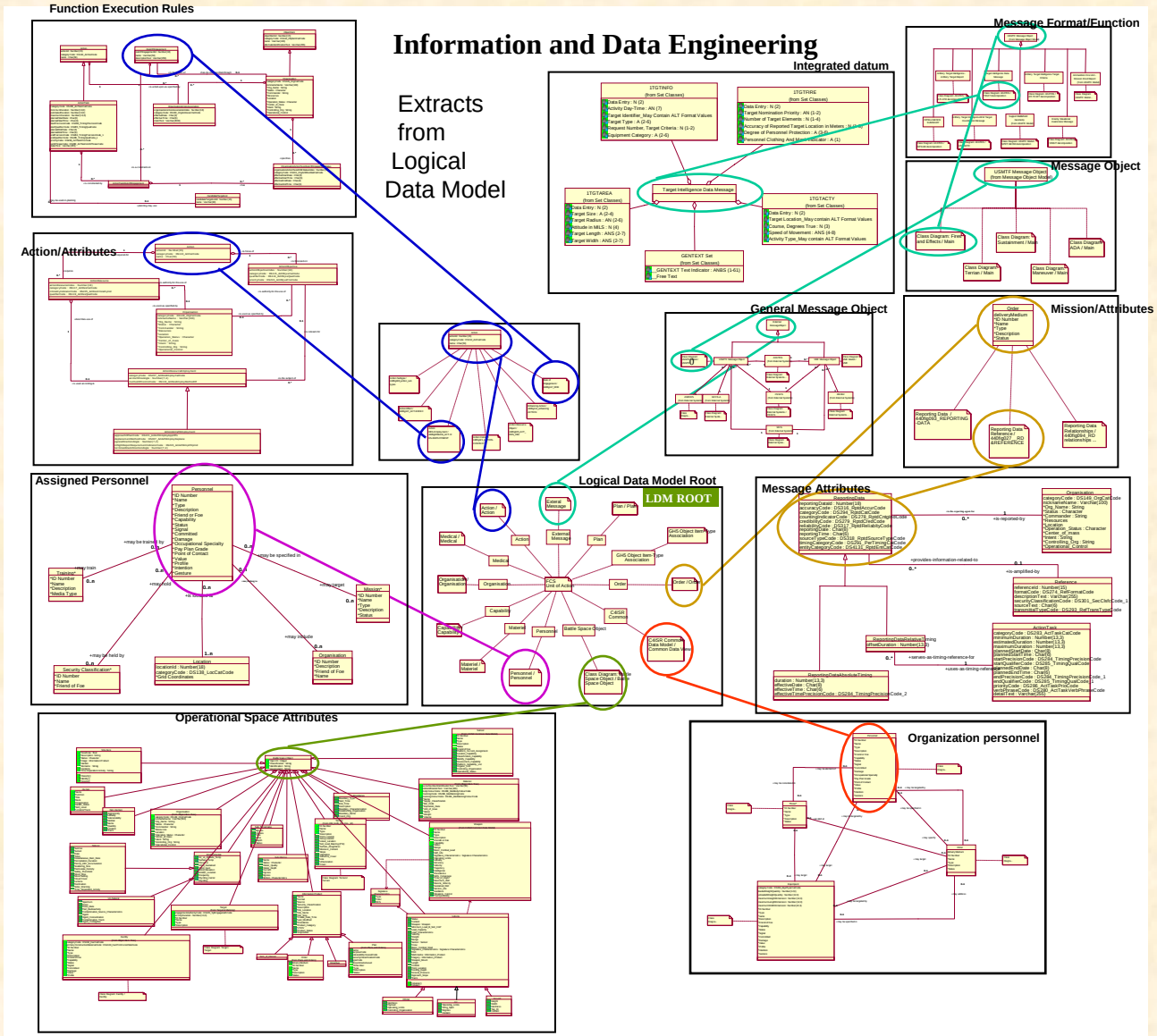
Architecture and Requirements Development Process



Design and Development Product Relationships



Data/Information Architecture Product Suite; Related



Why Now??

- Mostly because we are now living in the 6th generation of computer systems development and the 4th generation of information systems development. (We build them far better than we know how to use them)

- The power and capacity of our systems and applications are becoming complex (as opposed to just complicated) enough that without this we cannot understand the veracity of use or results.

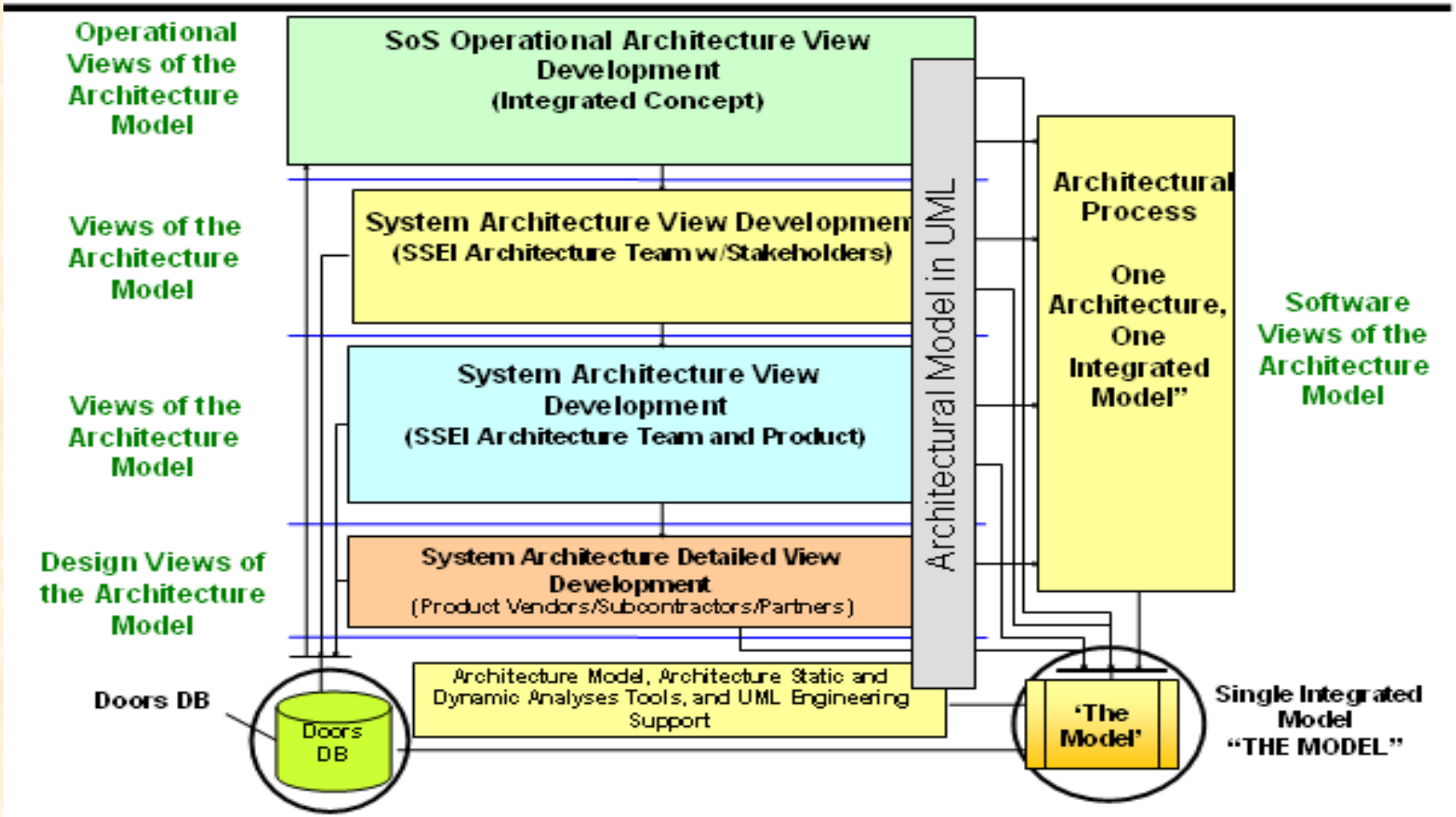
- It is today possible to specify an architecture framework, explain its pieces and parts, and actually apply knowledge to creating them.

- It is today possible to not only create them, but build them and apply them, automatically, and measure the benefits, empirically.

To What Purpose?

- **Foster collaboration more efficiently and effectively.**
- **Compel sponsors through empirically based road map production, almost automatically.**
- **Provide users with an understanding of how to fit in; disciplinarily, institutionally, and technically.**
- **Provide users with reuse capability on call, without need for justification and with faith in the value of that kind of redundancy.**
- **Provide public metrics of performance parameters for improvement (not a glass wall).**
- **Provide systems engineering support with less stress and cost.**
- **Allows research to focus on objectives not administration.**
- **Ensures the 'commons' are well understood and well nurtured.**
- **Systematically help determine what is needed; substantively and computationally.**
- **Ensures that constituent knowledge and expectations are acquired once, updated as the constituent dictates, and never lost.**

Integrated Architecture Process

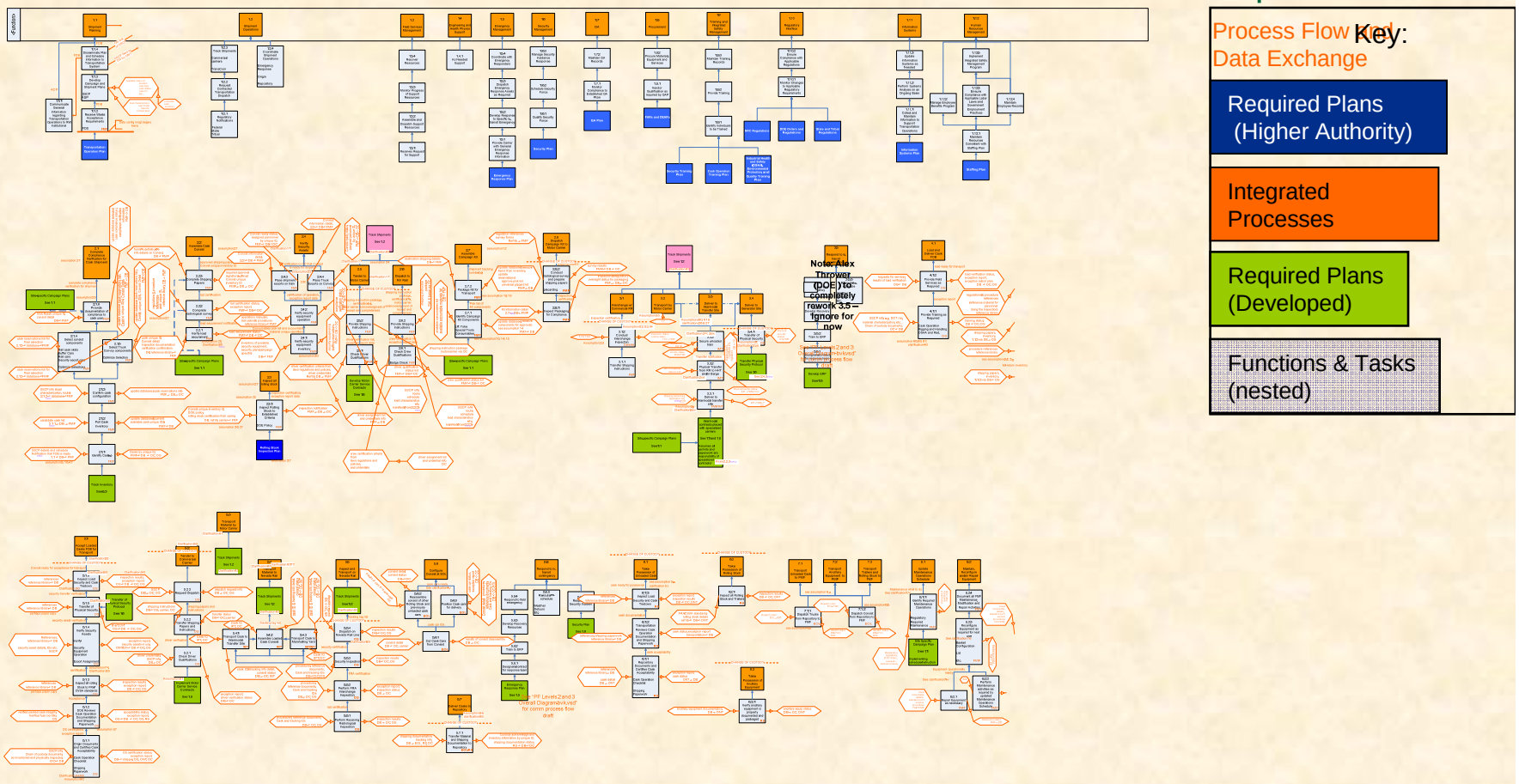


Backup

3rd Generation attempts at Architecture Development

DOE/ONT High Level Waste Transportation & Communications Systems:

WBS, Processes, Functions, Tasks, Process Flows, Functional Decomposition

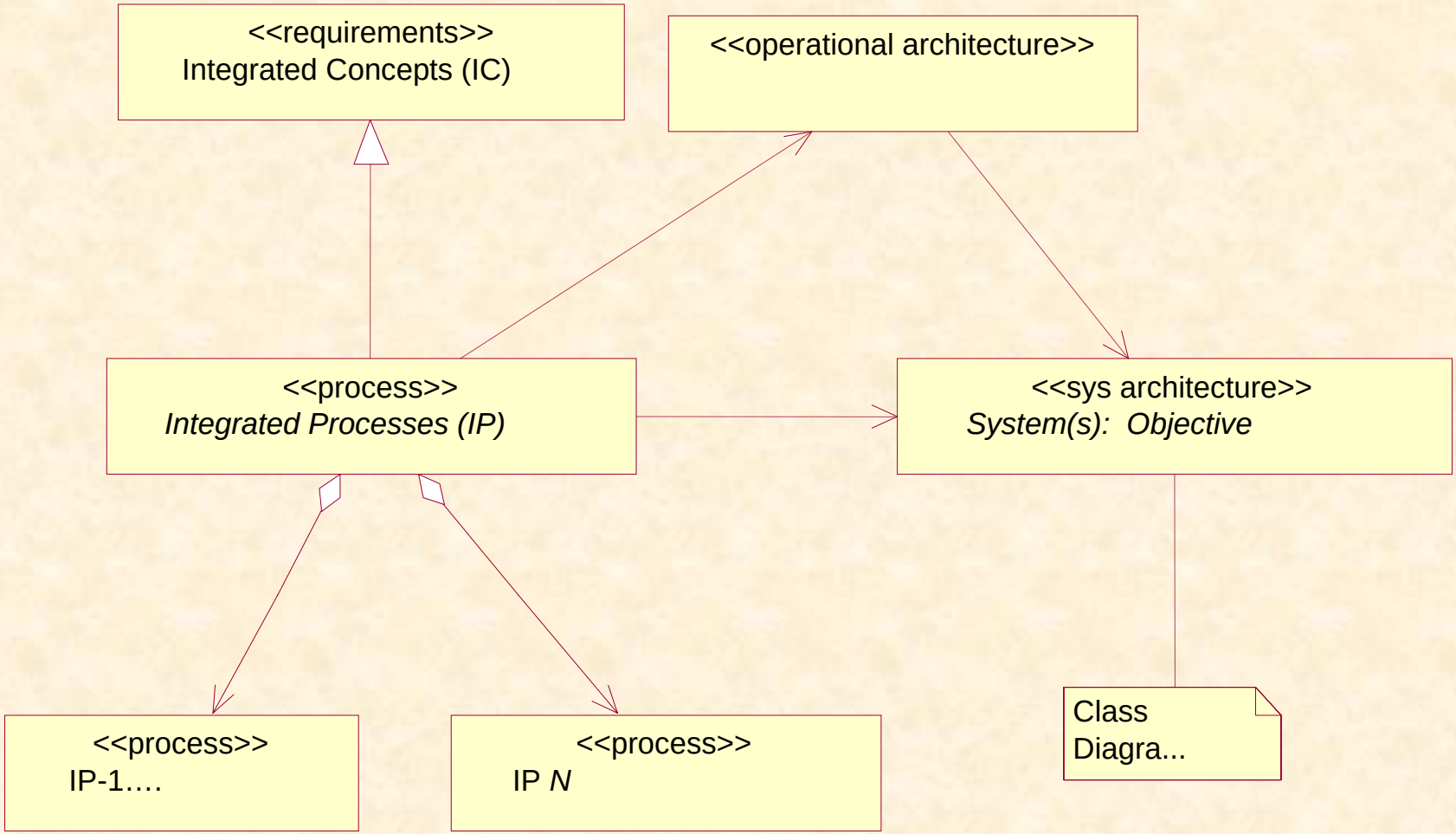


DODAF Architecture Products

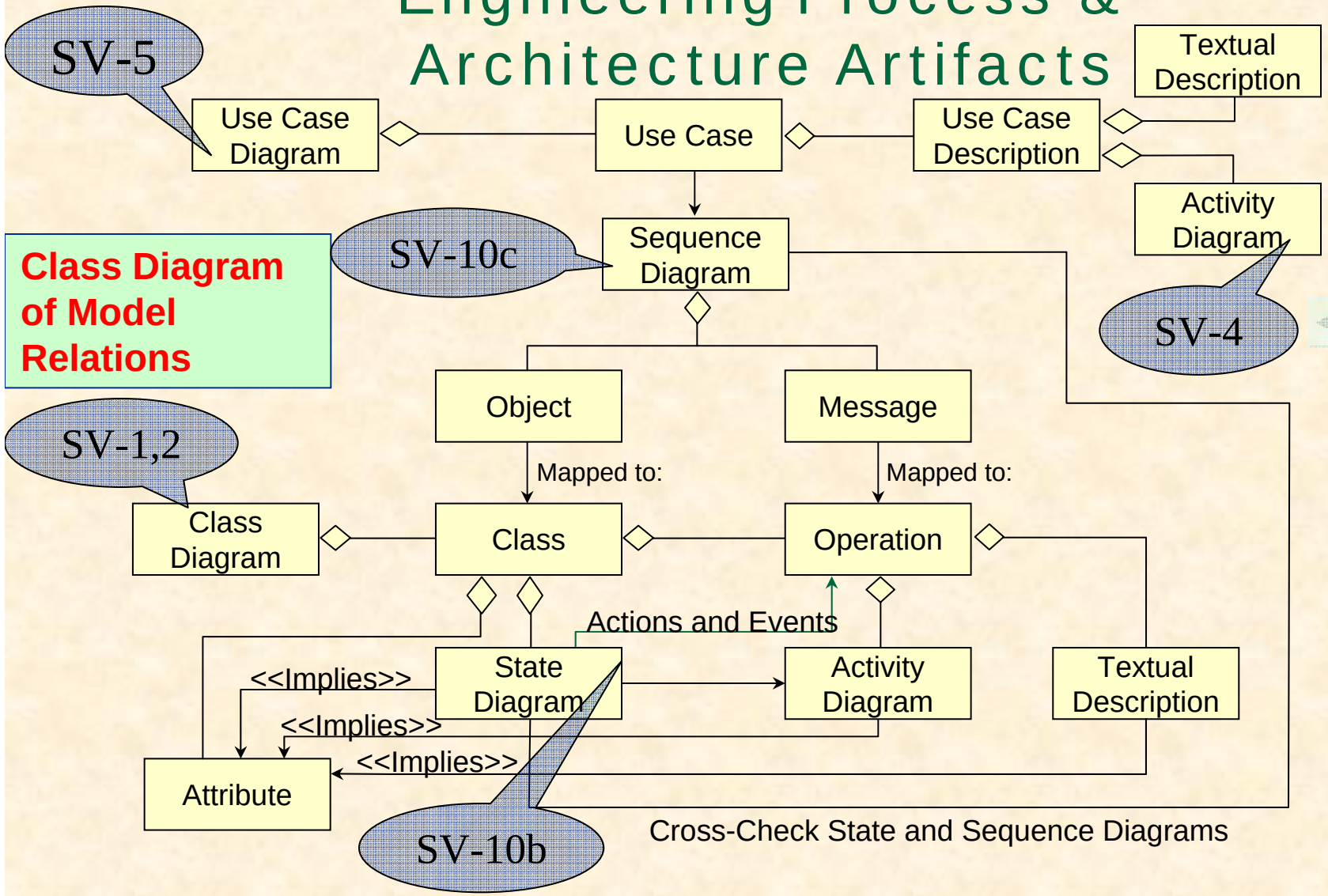
and Graded Example

Applicable View	Framework Product	Framework Product Name	DHS
All Views	AV-1	Overview and Summary Information	✓
All Views	AV-2	Integrated Dictionary	✓
Operational	OV-1	High-Level Operational Concept Graphic	✓
Operational	OV-2	Operational Node Connectivity Description	✓
Operational	OV-3	Operational Information Exchange Matrix	✓
Operational	OV-4	Organizational Relationships Chart	✓
Operational	OV-5	Operational Activity Model	
Operational	OV-6a, b, c	Operational Activity Sequence and Timing Descriptions	
Operational	OV-7	Systems-Systems Matrix Logical Data Model	
Systems	SV-1	Systems Interface Description	✓
Systems	SV-2	Systems Communications Description	✓
Systems	SV-3		✓
Systems	SV-4	Systems Functionality Description	✓
Systems	SV-5	Operational Activity to Systems Function Traceability Matrix	
Systems	SV-6	Systems Data Exchange Matrix	
Systems	SV-7	Systems Performance Parameters Matrix	
Systems	SV-8	Systems Evolution Description	✓
Systems	SV-9	Systems Technology Forecast	
Systems	SV-10a, b, c	Systems Functionality Sequence and Timing Descriptions	
Systems	SV-11	Physical Schema	
Technical	TV-1	Technical Standards Profile	✓
Technical	TV-2	Technical Standards Forecast	

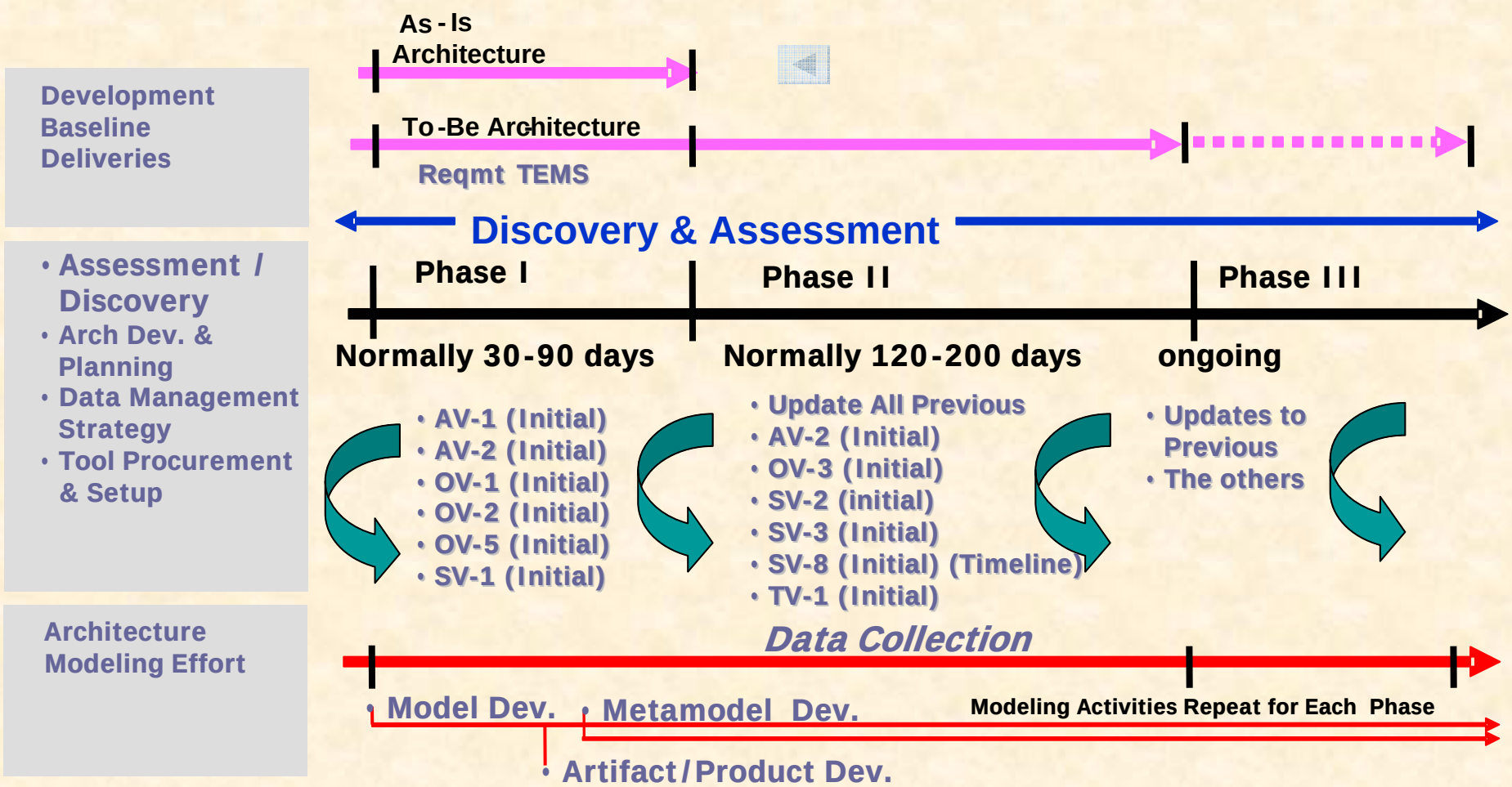
Use of Integrated Processes



Object Oriented Software Engineering Process & Architecture Artifacts



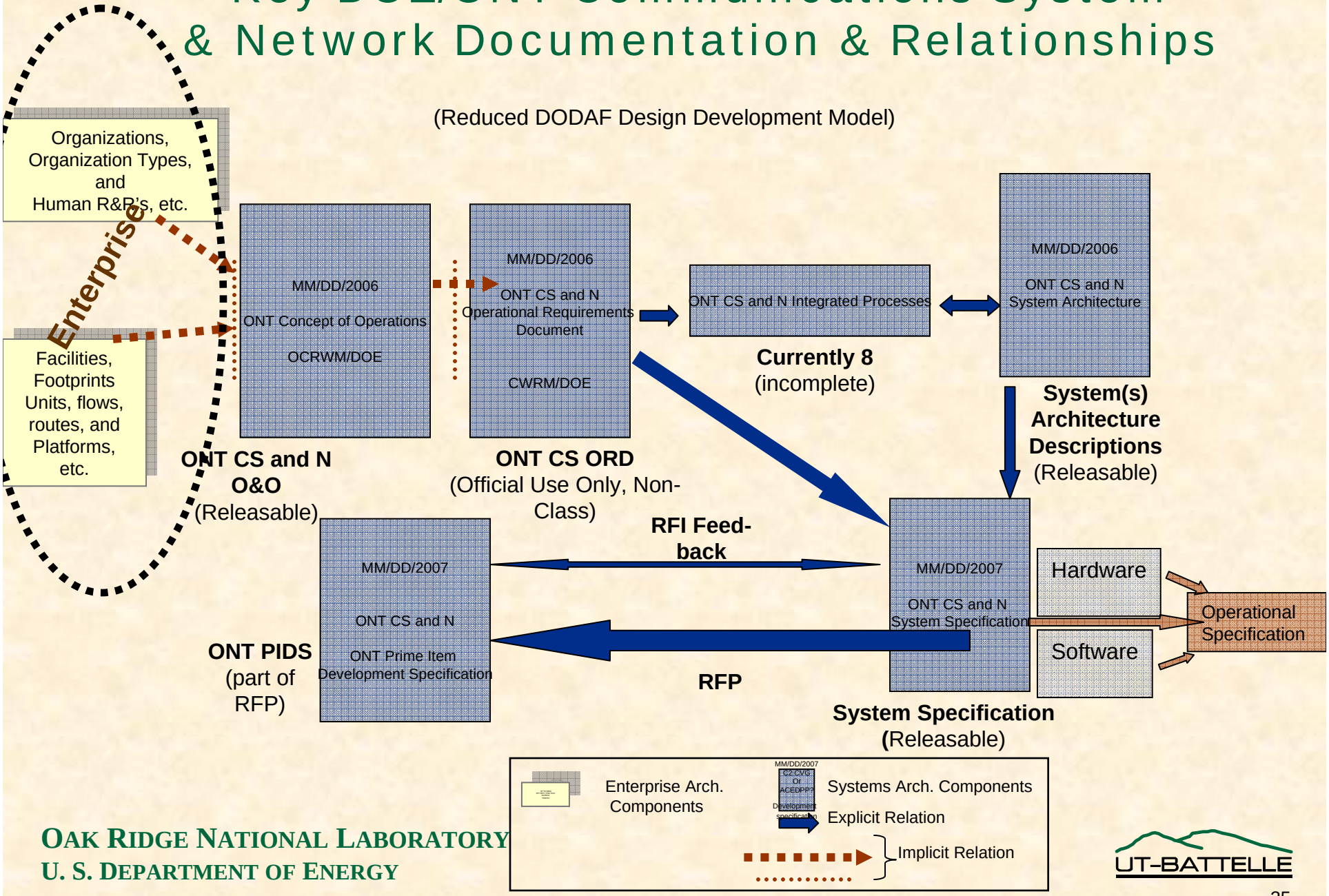
Enterprise Architecture Development Plan



NOTE: Architecture Artifacts/Maturity are dependant on availability of Data for their creation

Key DOE/ONT Communications System & Network Documentation & Relationships

(Reduced DODAF Design Development Model)



Trends

- **Beginning 6th Generation of Information Technology--remain in 4th Generation of Information Systems**
- **Formal Methods for effective SW development still 'immature'**
- **SW development remains too expensive and too uncertain [Boehm, Basili, etc.)**
- **Performance of SW is uncertain due to (e.g.):**
  - **mismatch between state of technology with systems maturity**
  - **knowledge & control emergence [Perow]**
  - **lack of adherence to formal methods**
  - **demands for mature functionality**
  - **knowledge and determinates of functional/operational integration**
- **The next leap in technology may not be as successful.**

‘Typical’ Evolution of a System(s) Architecture

1. **Propose a consistent set of plausible, expected capabilities.** (or issue ORD)
2. **Prepare operational, system, and technical requirements; showing the system(s), segments, subsystems, components, etc.**
3. **Define relationships among systems, subsystems, segments, components, etc.**
4. **Flow down requirements and interfaces to systems, subsystems, segments, components, etc.**
5. **Ascertain or realign systems, subsystems, segments, components to their requirements and performance constraints** (measures).
6. **Using documented relationships among systems, subsystems, segments, components, expected capabilities, etc, optimize performance characteristics** (electronic or otherwise).
7. **Objective System(s) architecture matures through repeated process application.**

Enterprise

Is any collection of organizations that have a common set of goals and/or a single bottom line.

Enterprise Architecture

Is the practice of applying a comprehensive and rigorous method for describing a current and/or future structure and behavior for an organization's;

- **processes,**
- **information systems,**
- **personnel, and**
- **organizational sub-units.**

Model Analysis Tools Characteristics

Architecture Analysis Tools

- **Profile Management Tool**
- **Logic Evaluation Tool**
- **Static Analysis Tool**
- **Discrete Event Simulation Integrated Development Environment**

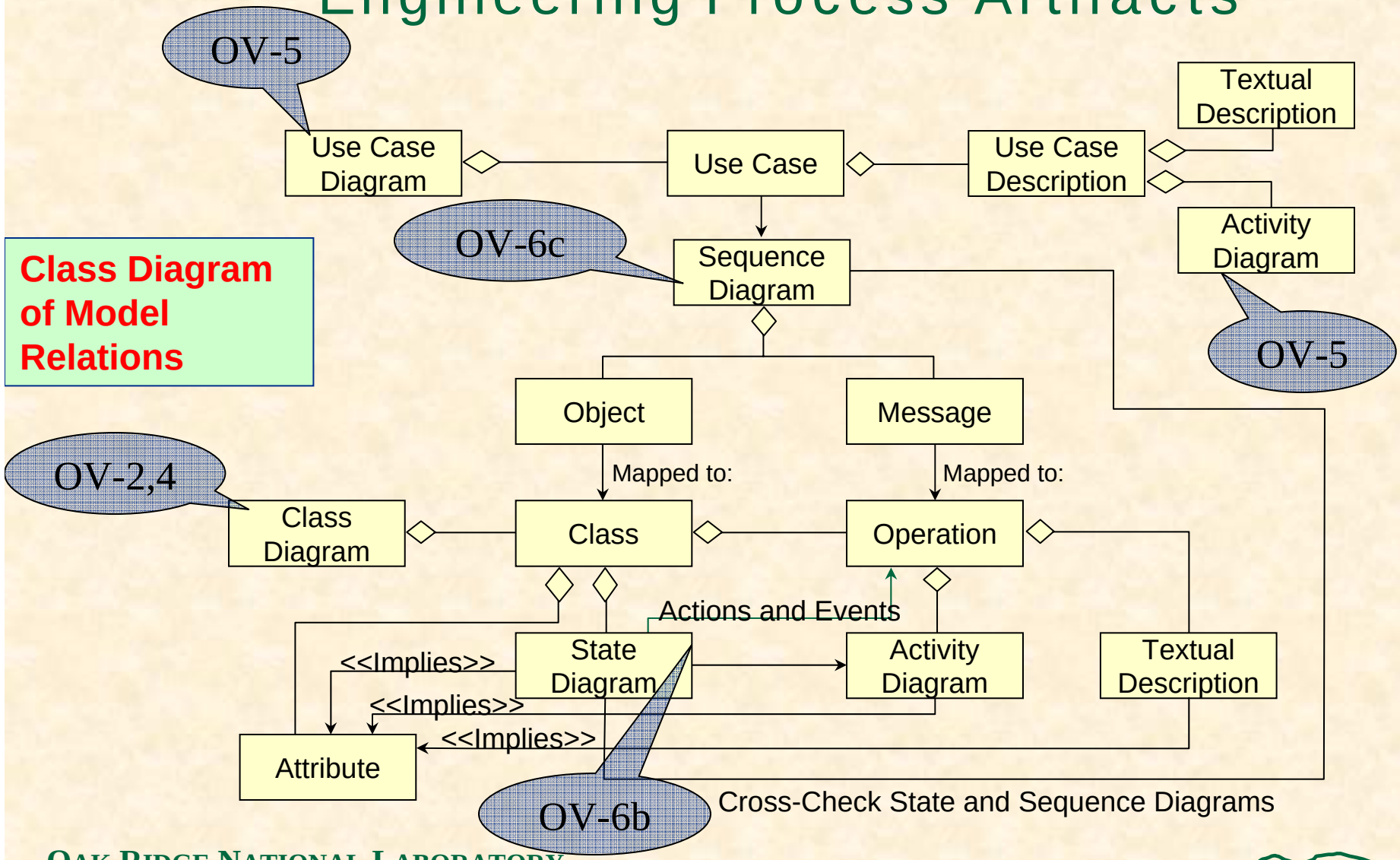
Model Status Tools

- **Use Case Status Tool**
- **Relationship Analyzer**

Model Analysis Tools

- **Style Checker Tool**
- **Activity and State Extraction Tool**

Object Oriented Software Engineering Process Artifacts



Outline

- Overview
- Architecture Introduction
- DoD Architecture Framework
- Mapping of DoDAF Products to UML
- System(s) Architecture
- One Single Integrated Model
- Objective System(s) Architecture
- Family of Systems
- Inter-Operational Systems
- Computing System Adjunct
- Integrated Processes
- Examples of Architectural Products
- Architecture Specification and Decomposition
- Summary and Future Trends

Combine Products and Practice Into Views

AV-1 – Overview and Summary: Scope, purpose, intended users, environment and analytical findings. Help to establish the overarching guidance for the system.
AV-2 – Integrated Dictionary: The architecture data repository; definitions of all terms used in all products. Establish standards for communication clarity across all products.
OV-1 – High-Level Operational Concept Graphic: Graphical and textual description of operational concept. Establishes major systems and communications modes between sub-systems.
OV-2 – Operational Node Connectivity Description: Connectivity & information exchange need-lines among nodes.
OV-3 – Operational Information Exchange Matrix: Information exchanged among nodes and attributes of exchange.
OV-4 – Organization Relationships chart: Depicts organizational, role or other relationships among organizations.
OV-5 – Operational Activity Model: Capabilities, operational activities, relationships among activities, inputs and outputs.
SV-1 – Systems Interface Description: Interconnections; within and among nodes, systems nodes, systems, and system items
SV-2 – Systems communications Description: Communications lay-downs, of related; nodes, systems, and system items
SV-3 – Systems-Systems Matrix: Relationships of interest among systems in a given architecture.
SV-4 – Systems Functionality Description: Functions performed by systems and the system data flows among system functions.
SV-8 – Systems Evolution Description: Planned incremental steps toward migrating a suite of systems to a more efficient suite in a future implementation.
TV-1 – Technical Standards Profile: Listing of standards that apply to Systems View elements in a given architecture.

	Platforms	Networks, Comm.	Software	Databases Formats	Business	User's, Human Factors	Law & Policy
As-Is	**TV 1**	**SV1 SV2**	**SV8**	**SV3**	**AV1 AV2 SV4**	**OV1 OV2 OV3 OV4 OV5**	***N/A***
To-Be	**TV 1**	**SV1 SV2**	**SV8**	**SV3**	**AV1 AV2 SV4**	**OV1 OV2 OV3 OV4 OV5**	***N/A***

Standard Practice Summary

- System(s)-specific architecture analysis and requirements determination conducted in context of strategic IT planning.
- Strategic plans often describe (colloquially):
  - Where are you now;
  - Where do you want to be;
  - How will you go there?
- Four architecture steps describe:
  - As-Is
  - To-Be
  - Planning
  - Final Architecture (How do you know you are done?)
- Consistent among all key factors of successful development are understood requirements for:

Platforms
Software
Law and Policy
Business Architecture
Networks and Communications
Databases and Data Formats
User Characteristics and Human Factors

"It is required"

The following policies and directives are applicable:

- **Executive Order 13011: Federal Information Technology (1996).**
- **Chief Information Officer Appointments**
- **OMB Circular A-11**
- **Clinger-Cohen Act of 1996 (40 USC 1401(3))**