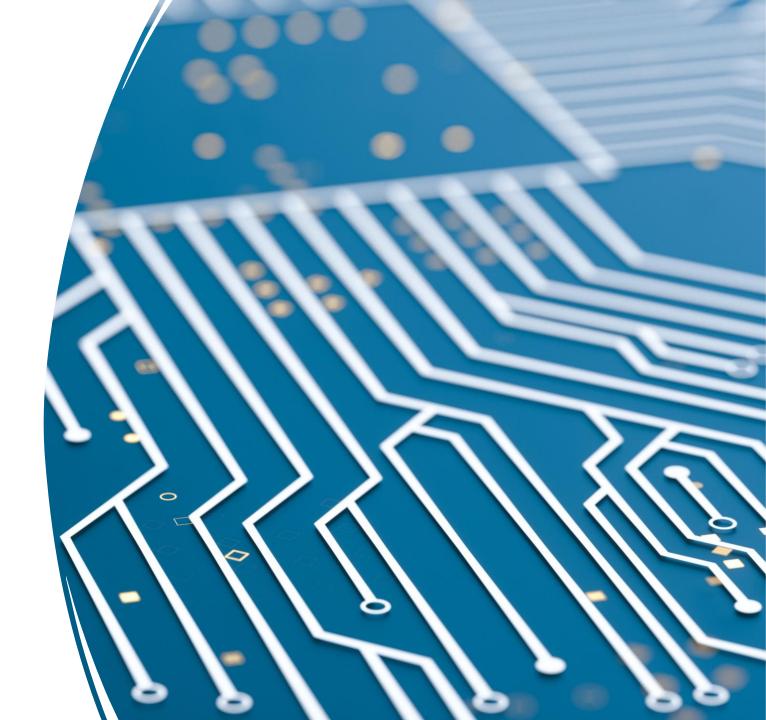


Who Am I

- Ph.D. Computer Science, Ph.D. Nanotechnology, D.Sc. Cybersecurity
- Four masters (Systems Engineering, Defense Studies, Education, Applied Computer Science)
- INCOSE certified systems engineer
- 44 books
- 27 patents
- Member of the American Society for Quality (Aviation, Space, and Defense Division)
- You can find movie at www.ChuckEasttom.com



Case 1

November 2012 The United States Air force had to cancel an Enterprise Resource Planning software project named "The Expeditionary Combat Support System" after it had cost approximately 1 Billion US Dollars but "failed to create any significant military capability". The costs are attributed to an "overwhelming" amount of additional custom coding and integration. It was determined that to complete the project would take another 1.1 billion dollars to get 1/4 of the original scope.

The U.S. Government Accountability Office released a report in March of 2012 that found many ongoing ERP projects by the nation's military are drastically behind schedule and over budget.

http://www.cio.com/article/721628/Air Force scraps massive ERP project after racking up 1 billion in co sts



Systems Engineering & Project Management

Program Management

- Planning
- Organizing
- Directing
- Monitoring
 - Maintaining insight or assessing
- Control

Common Areas

- Risk Mgt
- Config Mgt
- Performance
- Evaluation
- Mission
 Assurance

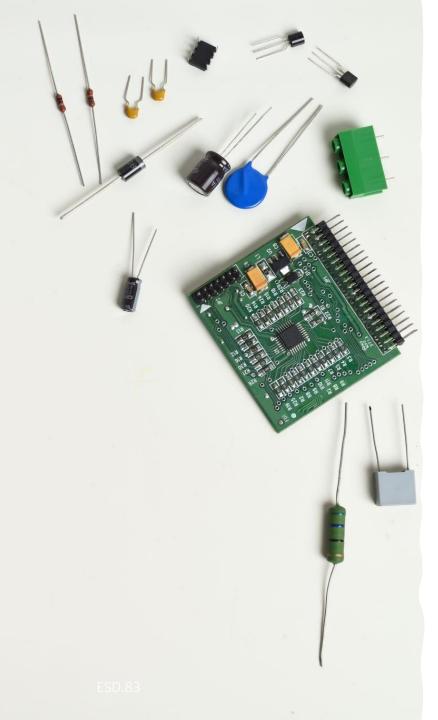
Systems Engineering

- Requirements & Functional Analysis and Allocation
- Design Synthesis
- Systems Analysis & Control (Balance)
- Integration &
- Verification

What is Systems Engineering?

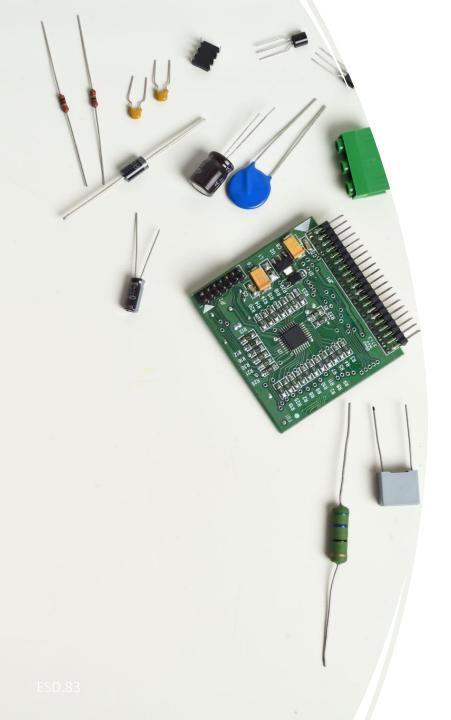
- Systems engineering is a robust approach to the design, creation, and operation of systems.
- The approach consists of:
 - identification and quantification of system goals
 - creation of alternative system *design* concepts
 - performance of *design* trades
 - selection and implementation of the best design
 - verification that the design is properly built and integrated, and
 - assessment of how well the system meets the goals
- This approach is iterative, with several increases in the resolution of the system baselines (which contain requirements, design details, verification plans and cost and performance estimates).
- NASA Systems Engineering Handbook SP-6105 (1995)





What is Systems Engineering?

- Systems Engineering is an *interdisciplinary approach* and means to enable the realization of successful systems.
- It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem:
 - Operations
 - Performance
 - Test
 - Manufacturing
 - Cost & Schedule
 - Training & Support
 - Disposal



What is Systems Engineering?

- Systems Engineering integrates all of the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation.
- Systems Engineering considers both the operational and the technical needs of all customers with the goal of *providing a quality product* that meets the user needs and delivers a military capability.



Why has Systems Engineering Emerged as A Distinct Discipline?

- The term itself was not formally used, nor was the importance of the concepts recognized, until after World War II.
- Complexity increased orders of magnitude with the creation of coupled mecho-digital systems, especially in defense (P-51 Mustang versus the Trident in 10 years)
- Creation of systems of systems, with users, acquisition, training, service, support, etc.

Original Reasons for Systems Engineering

- Systems of pieces built by different subsystem groups did not perform system functions
 - Often broke at the interfaces
- Problems emerged and desired properties did n
- when subsystems designed independently w integrated
- Managers and chief engineers tended to pay attention to the areas in which they were skilled
- Developed systems were not usable
- Cost overruns, schedule delays, performance problems



More Motivation for Systems Engineering

There is tremendous potential for wasted effort on large projects, since their development requires that many subsystems be developed in parallel.

Without a clear understanding of what must be done for each subsystem the development team runs the risk of inconsistent designs, conflicting interfaces or duplication of effort.

Systems engineering provides a systematic, disciplined approach to defining, for each member of the development team, what must be done for success.

NASA, DOD and Industry Call For More and Better Systems Engineers

All of the factors identified by NASA that contributed to program failure and significant cost overrun are systems engineering factors, e.g.,

- Inadequate requirements management
- Poor systems engineering processes
- Inadequate design analyses in early phases
- Inadequate systems-level risk management

Reference: NASA, Office of Program Analysis and Evaluation, Systems Engineering and Institutional Transitions Study, April 5, 2006. Reproduced in National Academies book - Building a Better NASA Workforce: Meeting the Workforce Needs for the National Vision for Space Exploration.

The Role of the System Engineer

Any engineer acts as a systems engineer when responsible for the design and implementation of a total system.

The difference with "traditional engineering" lies primarily in the greater emphasis on defining goals, the creative generation of alternative designs, the evaluation of alternative designs, and the coordination and control of the diverse tasks that are necessary to create a complex system.

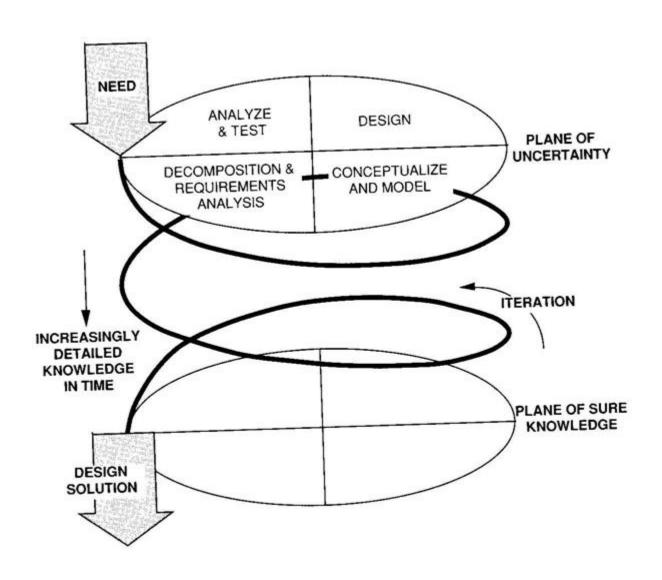
The role of Systems Engineer is one of *Manager* that utilizes a structured value delivery process



The Systems Engineering Process

- The major steps in the completion of a typical systems engineering project are the following: (1) problem statement; (2) identification of objectives; (3) generation of alternatives; (4) analysis of these alternatives; (5) selection of one of them; (6) creation of the system, and, finally, (7) operation.
- Some examples of Systems Engineering Process activities are:
 - Defining needs, operational concept, and requirements
 - Functional analysis, decomposition, and allocation
 - System modeling, systems analysis, and tradeoff studies
 - Requirements allocation, traceability, and control
 - Prototyping, Integration, and Verification
 - System Engineering Product and Process control
 - Configuration and Data Management
 - Risk Management approaches
 - Engineering technical reviews and their purposes

Systems
Engineering
Methodologies



Overview

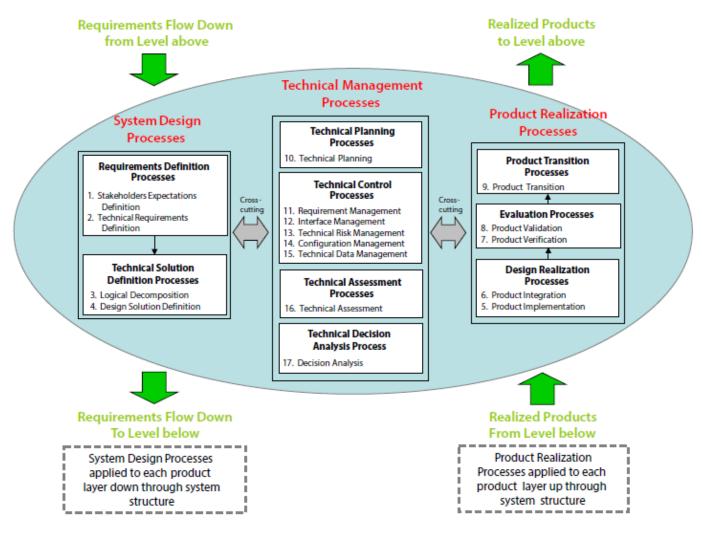


FIGURE 2.1-1 The Systems Engineering Engine (NPR 7123.1)

-NASA Systems Engineering Handbook

AS9100 is a widely adopted and standardized quality management system

Systems engineering and AS9100

TABLE 2.1-1 Alignment of the 17 SE Processes to AS9100

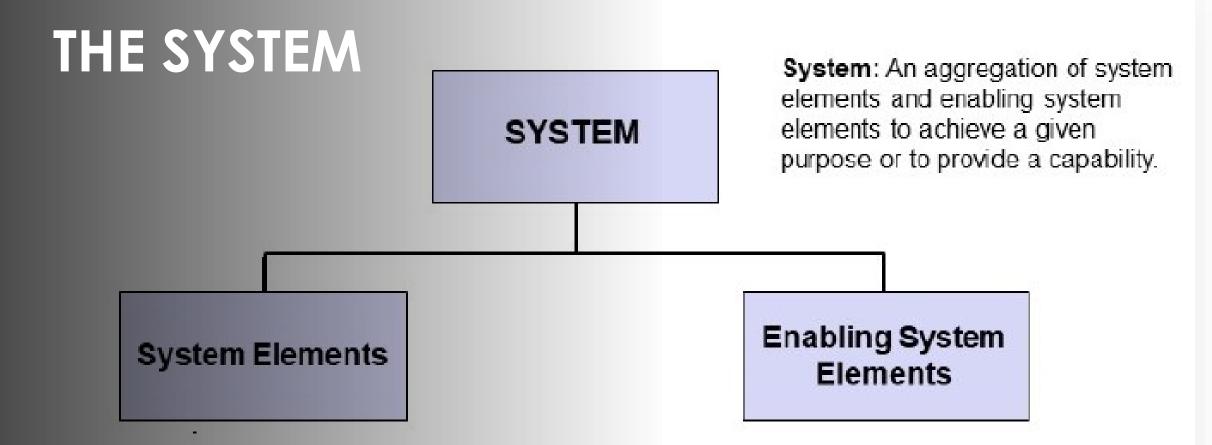
SE Process	AS9100 Requirement
Stakeholder Expectations	Customer Requirements
Technical Requirements Definition	Planning of Product Realization
Logical Decomposition	Design and Development Input
Design Solution Definition	Design and Development Output
Product Implementation	Control of Production
Product Integration	Control of Production
Product Verification	Verification
Product Validation	Validation
Product Transition	Control of Work Transfers; Post Delivery Support, Preservation of Product
Technical Planning	Planning of Product Realization; Review of Requirements; Measurement, Analysis and Improvement
Requirements Management	Design and Development Planning; Purchasing
Interface Management	Configuration Management
Technical Risk Management	Risk Management
Configuration Management	Configuration Management; Identification and Traceability; Control of Nonconforming Product
Technical Data Management	Control of Documents; Control of Records; Control of Design and Development Changes
Technical Assessment	Design and Development Review
Decision Analysis	Measurement, Analysis and Improvement; Analysis of Data

What is a System?

Simply stated, a system is an integrated composite of people, products, and processes that provide a capability to satisfy a stated need or objectives.







System Elements: Also referred to as configuration items, subsystems, segments, components, assemblies, or parts.

Enabling System Elements: Provide the means for putting a capability into service, keeping it in service, or ending its service, e.g., processes or products used to enable system development, test, production, training, deployment, support, and disposal.



What is Systems Engineering?

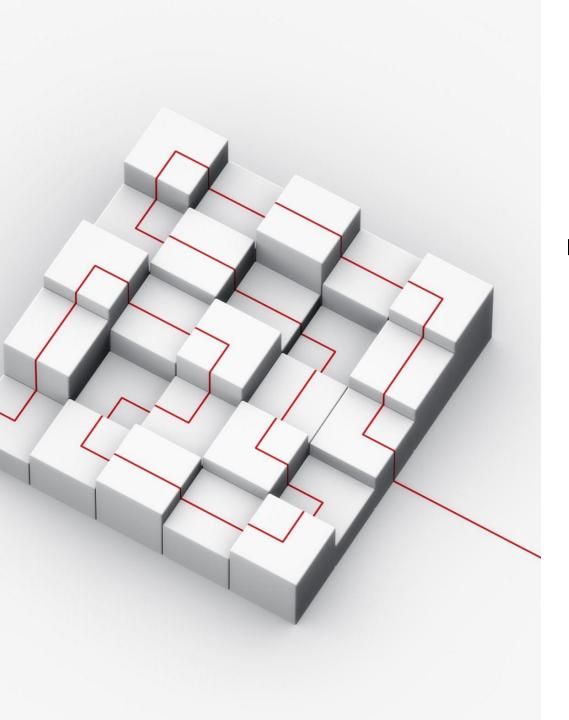
"Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems."

INCOSE Handbook



In simple terms, the systems engineering approach consists of:

- Identification and quantification of system goals,
- Creation of alternative system design concepts,
- Performance of design trades,
- Selection and implementation of the best design,
- Verification that the design is properly built and integrated, and
- Post implementation assessment of how well the system meets (or met) the goals

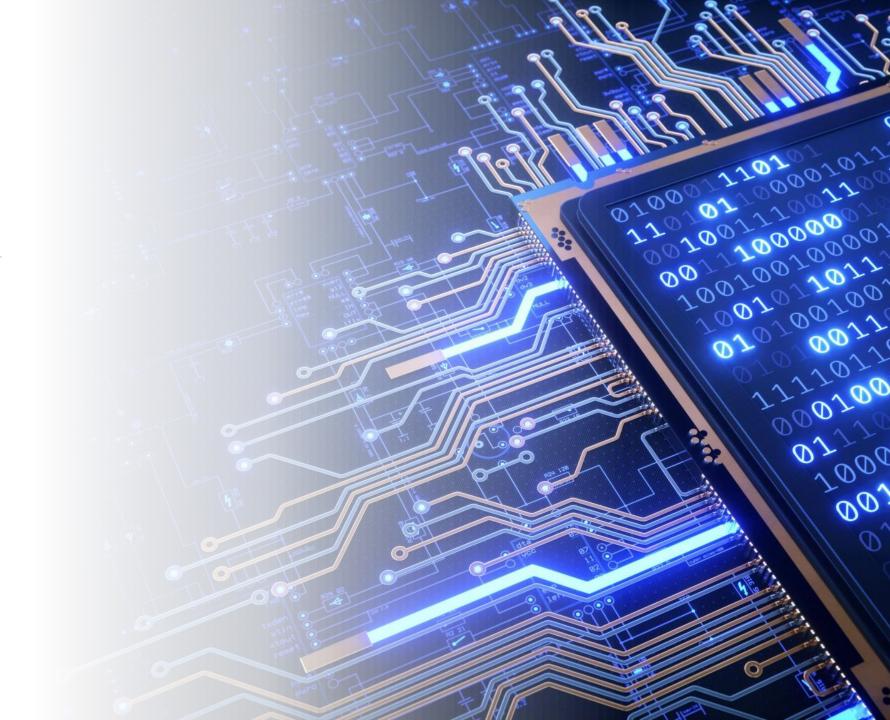


In simple terms, the systems engineering approach consists of:

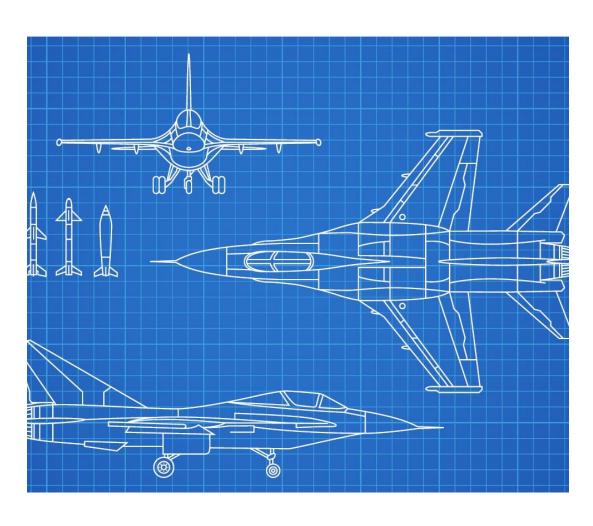
- Identification and quantification of system goals,
- Creation of alternative system design concepts,
- Performance of design trades,
- Selection and implementation of the best design,
- Verification that the design is properly built and integrated, and
- Post implementation assessment of how well the system meets (or met) the goals

"Engineering of Systems"

Anyone involved in engineering a system should exercise good systems engineering practices.



Systems Engineering Terms



- LORA Level Of Repair Analysis
- LR Launch Reliability
- LRM Line Replaceable Module
- LRU Line Replaceable Unit
- MAJCO Major Command
- MAP Mission Area Plan
- MBE Model Based Engineering
- MBIT Maintenance Built-in Test
- MCMT Mean Corrective Maintenance Time
- MCTI Mean Cost To Isolate
- MCTR Mean Cost To Repair / Mean Cost To Replace
- MDT Mean Downtime
- MEFL Mission Essential Functions List
- MESL Minimum-Essential Subsystem List
- MLH/AH Maintenance Labor Hours per Active

Hour

MMH/FH Maintenance Man-Hours per Flying Hour

Systems Engineering Terms

- MNS Mission Needs Statement
- MP Mission Profile
- MP/U Maintenance Personnel per Operational Unit
- MRS Mobility Requirements Study
- MSMT Mean Scheduled Maintenance Time
- MTBCF Mean Time Between Critical Failure
- MTBDE Mean Time Between Downing Event
- MTBF Mean Time Between Failures
- MTBFA Mean Time Between False Alarms
- MTBM Mean Time Between Maintenance
- MTBMA Mean Time Between Maintenance

Actions

MTBPM Mean Time Between Preventative

Maintenance

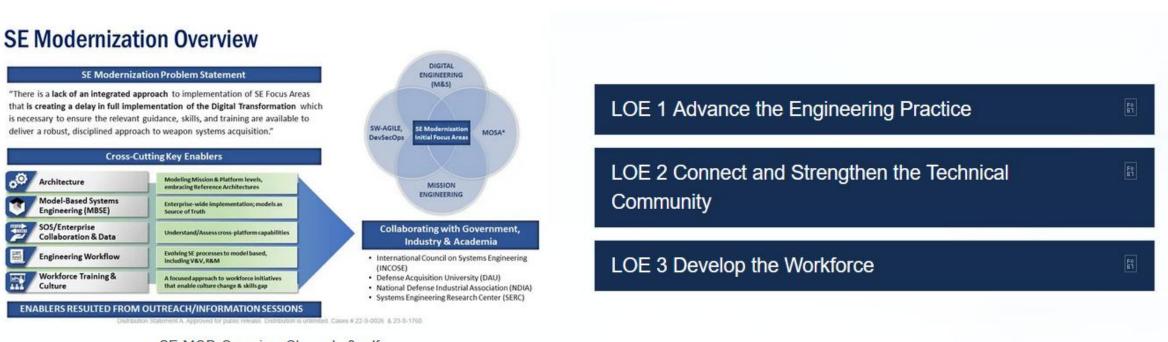
- MTBSA Mean Time Between System Aborts
- MTBR Mean Time Between Removal
- MTBUM Mean Time Between Unscheduled

Maintenance

Systems Engineering Terms

- MTBSM Mean Time Between Scheduled Maintenance
- MTD Maintenance Training Device
- MTTF Mean Time to Failure
- MTTI Mean Time to Isolate
- MTTR Mean Time to Repair / Mean Time to Replace
- MTTRF Mean Time to Restore Function
- MTTRS Mean Time to Restore System
- NMC Not Mission Capable
- OM Opportunistic Maintenance
- OMS Operational Mission Summary
- ORD Operational Requirements Document
- PdM Predictive Maintenance
- PoF Probability of Failure / Physics of Failure
- PRA Probabilistic Risk Assessment
- PBRA Probabilistic-Based Risk Assessment
- WRA Weapon Replaceable Assembly
- WSR Weapon System Reliability

DoD Systems Engineering Modernization



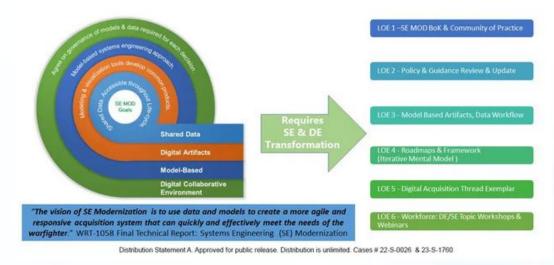
SE-MOD-Overview-Cleared-v3.pdf

https://www.cto.mil/sea/se/

DoD Systems Engineering Modernization



SE Modernization Goals & Lines of Effort (LOEs)



SE-MOD-Goals-LOEs-Cleared-v3.pdf

+

0

Systems Theory

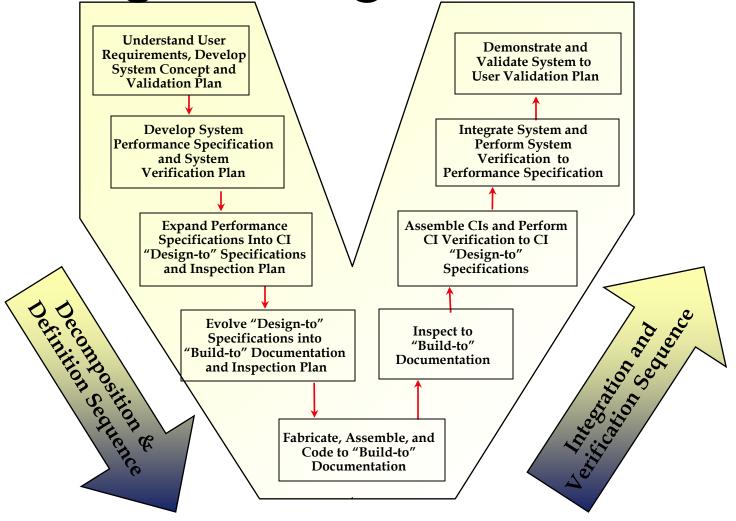
Systems engineering is, of course, related to systems theory. Systems Theory is an interdisciplinary framework for understanding, analyzing, and designing complex entities made up of interrelated parts. Instead of studying components in isolation, it emphasizes the relationships, interactions, and dynamics among parts within a whole system. Systems theory originated in biology but has since expanded to engineering, information science, social sciences, management, and beyond. Ludwig von Bertalanffy (1901–1972), a biologist, is considered the father of General Systems Theory (GST).

+ Systems Theory

Core Principles of Systems Theory

- Holism (The Whole is Greater than the Sum of its Parts)
 - Systems cannot be understood by examining components alone.
 - Example: A human body's health emerges from the interactions of organs, not just the organs themselves.
- Interdependence & Interaction
 - System elements interact and depend on one another. A change in one part affects the rest.
- Boundaries
 - A system is defined by what is inside vs. outside (its environment). Boundaries may be physical or conceptual.
- Inputs, Processes, Outputs, Feedback
 - Every system takes inputs from the environment, processes them, produces outputs, and uses feedback to adapt.
- Open vs. Closed Systems
 - Open systems exchange energy, matter, or information with the environment.
 - Closed systems have minimal exchange (mostly theoretical in practice).
- Homeostasis and Equilibrium
 - Systems tend to self-regulate and maintain stability through feedback loops.
- Hierarchy and Subsystems
 - Systems are often nested: subsystems within systems, and systems within larger systems.
- Emergence
 - New properties emerge at the system level that cannot be reduced to the properties of individual parts.

Systems Engineering Process "V"



DoD Systems Engineering V

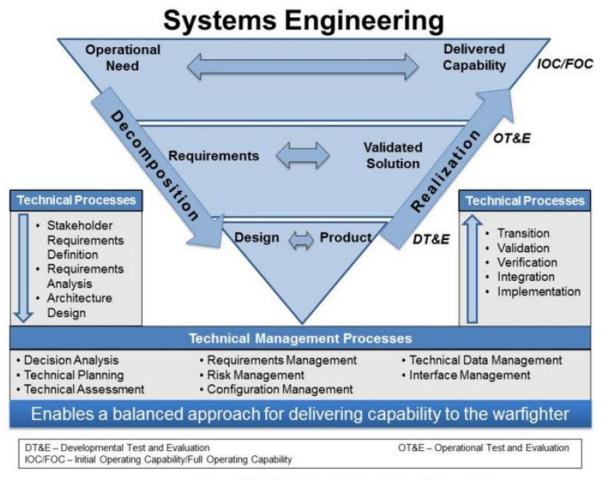


Figure 1-2. Systems Engineering Processes

DoD SE Policies

Table 1-1. Systems Engineering-Related Policy

Systems Engineering–Related Policy	Source
DoD Directive 5000.01, The Defense Acquisition System	Office of the Under Secretary of Defense for Acquisition and Sustainment, September 9, 2020
DoD Instruction 5000.02, Operation of the Adaptive Acquisition Framework	Office of the Under Secretary of Defense for Acquisition and Sustainment, January 23, 2020
DoD Directive 5137.02, Under Secretary of Defense for Research and Engineering (USD(R&E))	Office of the Chief Management Officer of the Department of Defense, July 15, 2020
DoD Instruction 5000.88, Engineering of Defense Systems	Office of the Under Secretary of Defense for Research and Engineering, November 18, 2020



The Role of the System Engineer

- Any engineer acts as a systems engineer when responsible for the design and implementation of a total system.
- The difference with "traditional engineering" lies primarily in the greater emphasis on defining goals, the creative generation of alternative designs, the evaluation of alternative designs, and the coordination and control of the diverse tasks that are necessary to create a complex system.
- The role of Systems Engineer is one of *Manager* that utilizes a structured value delivery process



The Systems Engineering Process

- The major steps in the completion of a typical systems engineering project are the following: (1) problem statement; (2) identification of objectives; (3) generation of alternatives; (4) analysis of these alternatives; (5) selection of one of them; (6) creation of the system, and, finally, (7) operation.
- Some examples of Systems Engineering Process activities are:
 - Defining needs, operational concept, and requirements
 - Functional analysis, decomposition, and allocation
 - System modeling, systems analysis, and tradeoff studies
 - Requirements allocation, traceability, and control
 - Prototyping, Integration, and Verification
 - System Engineering Product and Process control
 - Configuration and Data Management
 - Risk Management approaches
 - Engineering technical reviews and their purposes

SoS, SOI, MS, & SS



SoS - Systems that interact with each, but were not all intentionally designed to work together, may exhibit unexpected behaviors upon interaction.



SOI – System of interest, the system you are currently focused on.



MS – Mission system, the primary system to meet the need.



SS – Support system, a system that aids the MS.

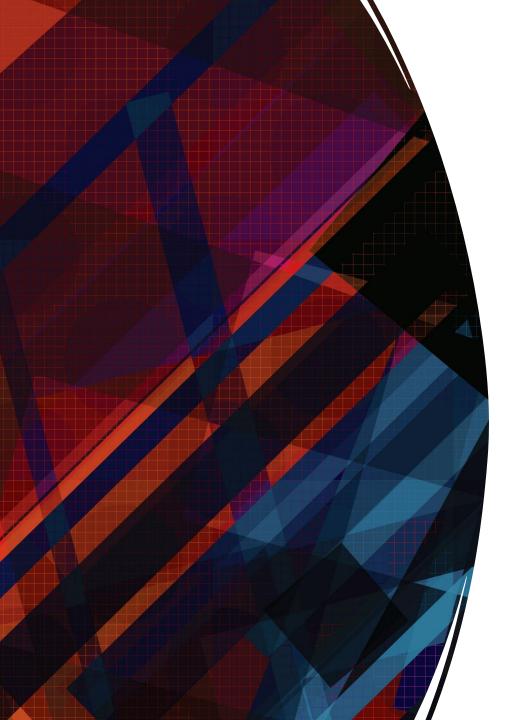


Operations Concept

- An Operations Concept is a vision (in general terms) for what the system is, and a description of how the system will be used.
- An Operations Concept consists of a set of scenarios describing how the system will be used during all of its operational phases.
- The scenarios are often accompanied by illustrations of the system operations.
- An Operations Concept:
 - Serves as a validation reference for the design throughout the life cycle
 - Describes how the design can accomplish the mission described by the objectives
 - Key to defining all the requirements
 - Evolves into the flight operations plan later in the life cycle

OpsCon vs ConOps

- ANSI/ AIAA G-043A- 2012 states that the terms "concept of operations" and "operational concept" are often used interchangeably but notes that an important distinction exists in that each has a separate purpose and is used to meet different ends.
- ISO/ IEC/ IEEE 29148 describes the ConOps as: The ConOps, at the organization level, addresses the leadership's intended way of operating the organization. It may refer to the use of one or more systems, as black boxes, to forward the
- ISO/ IEC/ IEEE 29148 describes the OpsCon as A System Operational Concept (OpsCon) document describes what the system will do (not how it will do it) and why (rationale). An OpsCon is a user-oriented document that describes system characteristics of the to-be-delivered system from the user's viewpoint.



OpsCon vs ConOps

- Concept of Operations: A verbal and graphic statement, in broad outline, of an enterprise's assumptions or intent regarding an operation or series of operations. The concept of operations frequently is embodied in long-range strategic plans and annual operational plans. In the latter case, the concept of operations in the plan covers a series of connected operations to be carried out simultaneously or in succession. The concept is designed to give an overall picture of the enterprise operations. It is also called the CONOPS
- Short definition: A document describing the characteristics of a proposed system from the viewpoint of the people who will use it.
- Operational Concept: A verbal and graphic statement of an enterprise's assumptions or intent regarding an operation or series of operations of a system or a related set of systems. The operational concept is frequently developed as part of a system development or acquisition program. The operational concept is designed to give an overall picture of the operations using one or more specific systems, or set of related systems, in the enterprise's operational environment from the users' and operators' perspective. It is also called the OpsCon. It is defined in an Operational Concept Document

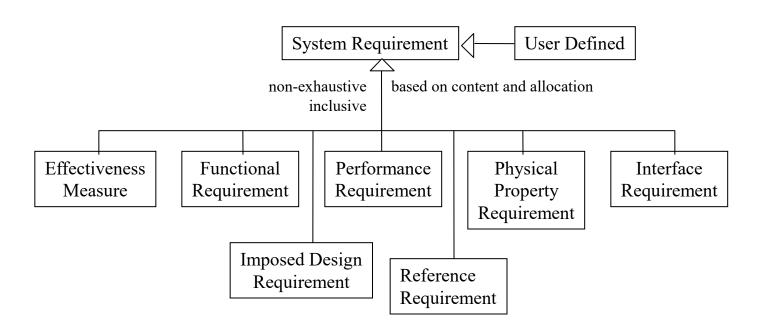


Decomposition

Many types of decomposition

- Requirements Decomposition
- Functional Decomposition
 - Functional Architecture
- Physical Decomposition
 - Physical Architecture
- Operational Architecture
 - Allocates functions to physical subsystems
 - Provides complete description of the system design
 - Integrates the requirements decomposition with the functional and physical architectures

Decomposition (Continued)



Systems Analysis and Design

- Models are the language of the designer.
- Models are representations of the system-to-be-built or as-built.
- Models are a vehicle for communications with various stakeholders.
- Models allow reasoning about characteristics of the real system.
- Models can be used for verification by analysis.
- All models must themselves be verified.



System Model Restraining Factors

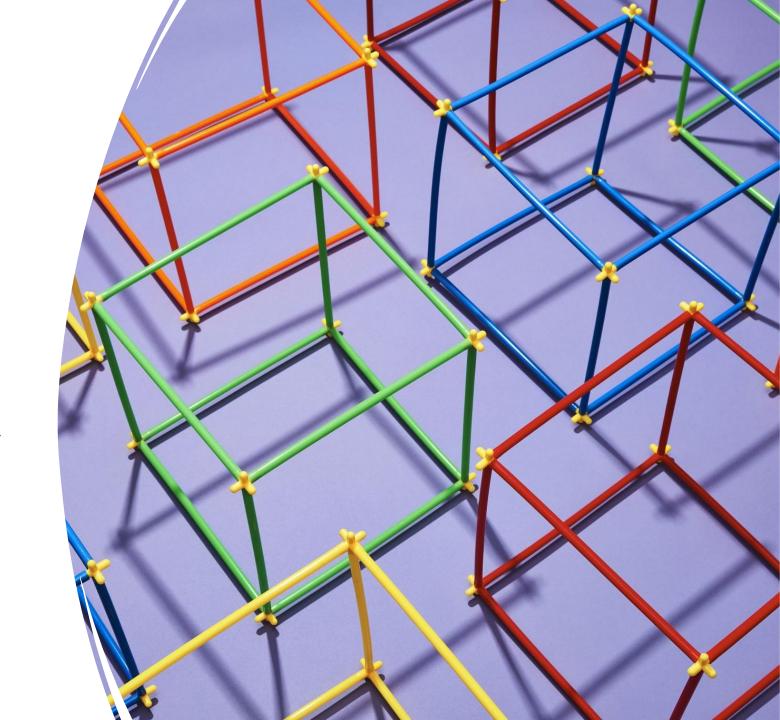
- Assumptions
- Simplifications
- Limitations
- Constraints
- Preferences

Architectural views

What views or perspectives are useful when designing and documenting a system's architecture?

What notations should be used for describing architectural models?

Each architectural model only shows one view or perspective of the system. It might show how a system is decomposed into modules, how the run-time processes interact or the different ways in which system components are distributed across a network. For both design and documentation, you usually need to present multiple views of the software architecture.



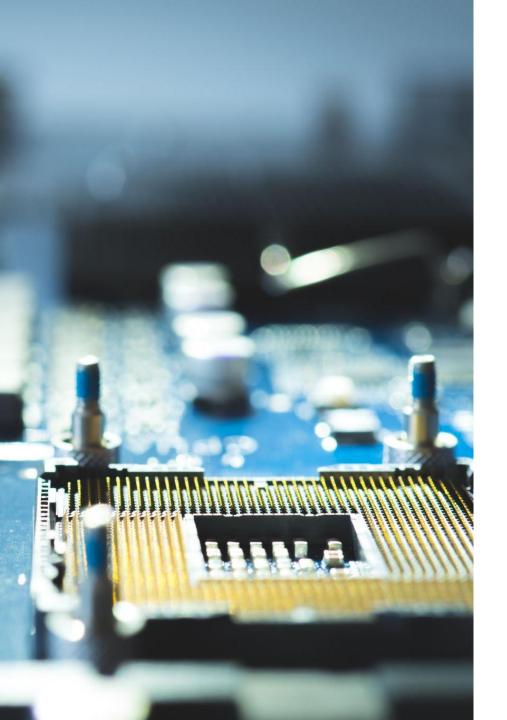
Architectural patterns

Patterns are a means of representing, sharing and reusing knowledge.

An architectural pattern is a stylized description of good design practice, which has been tried and tested in different environments.

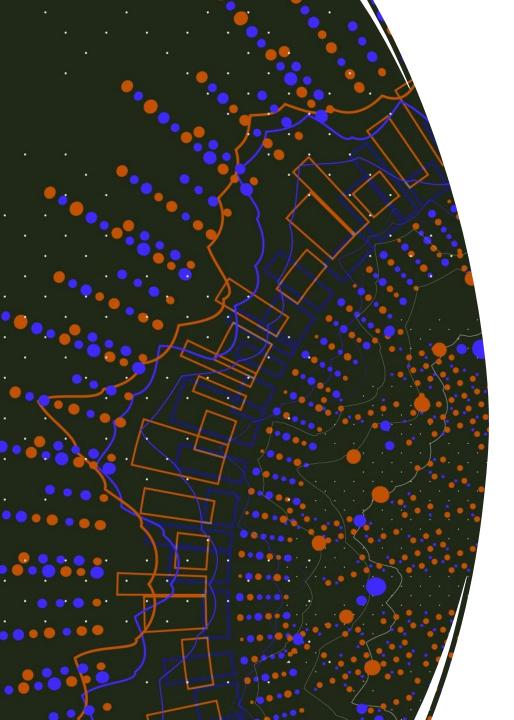
Patterns should include information about when they are and when the are not useful.

Patterns may be represented using tabular and graphical descriptions.



System Integration

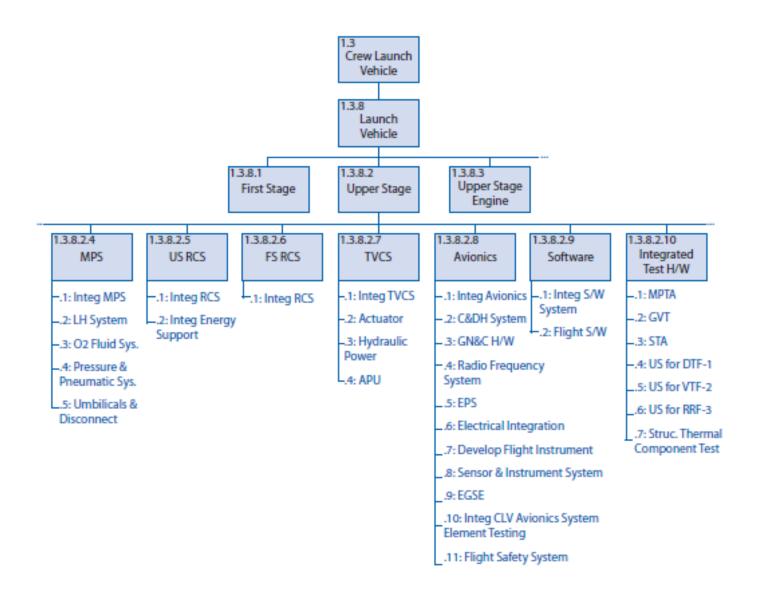
- Integration is the process of assembling the system from components.
- Integration begins with the elementary pieces or configuration items (Cl's) of the system.
- After each CI is tested, components comprising multiple CI's are tested.
- This process continues until the entire system is assembled and tested.
- Interface Specifications and Interface Control are critical to a successful system integration.

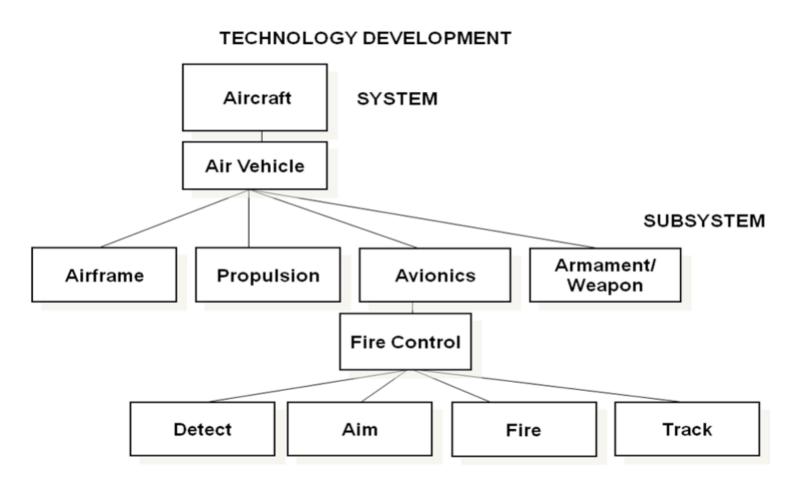


Work and Resource Management

- A Work Breakdown Structure (WBS) is a hierarchical breakdown of the work necessary to complete a project.
- The WBS should be a product-based, hierarchical division of deliverable items and associated services.
- The WBS should contain the Product Breakdown Structure (PBS).
- At the lowest level are products such as hardware items, software items, and information items (documents, databases, etc.) for which there is a cognizant engineer or manager.
- A project WBS should be carried down to the cost account level appropriate to the risks to be managed.

Work Breakdown Structure (WBS)





WBS

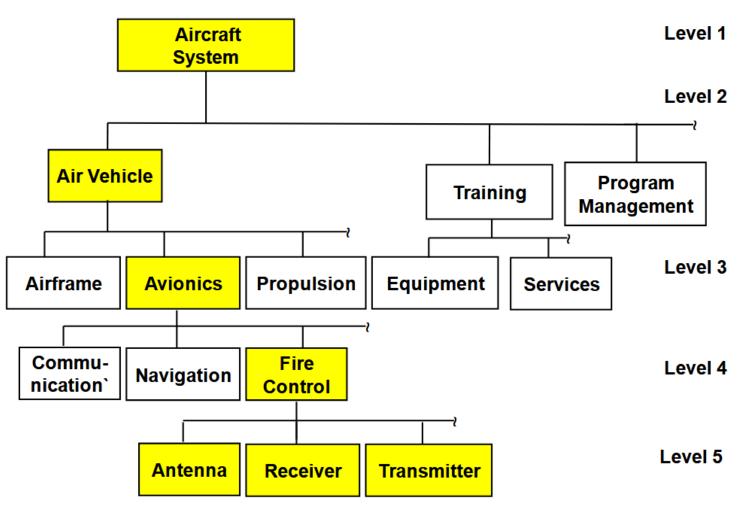


FIGURE 5. Program WBS Description

Maintainability

Maintainability is that system design characteristic associated with the ease and rapidity with which the system can be retained in operational status, or safely and economically restored to operational status following a failure.



Verification

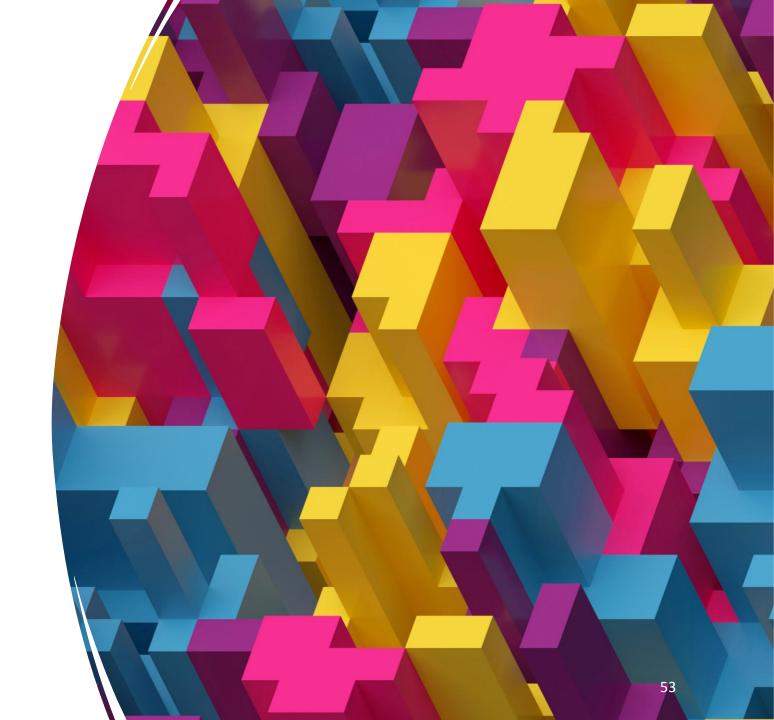
- Verification: "Did I build the System Right?"
- Each requirement must be verified
- Verification Methods: Test, Analysis, Inspection and Demonstration
- Rule #1: "Test wherever possible"
 - Perform Analysis and Inspection, where Test is not possible
 - Pay careful attention to validity of simulators and models
- Rule #2: "Test the way you use, use the way you test"
 - Identify what is not tested in flight configuration
 - Careful review to assure items are properly verified by a combination of Analysis, Inspection or Test.
 - Review of the assumptions and interfaces of element verified in pieces
 - Attention to validity of simulators and simulations
 - Careful review to assure these items are properly verified by a combination of Analysis, Inspection or Test.
- Rule #3: "Test the system end-to-end"
 - Carefully review the assumptions and interfaces of any elements verified in pieces
- Rule #4: "Verify Off-Nominal Conditions"
 - Verify Redundancy and Graceful Degradation Modes along with On Board Fault Protection and Ground Contingency Procedures
 - Stress Testing and Negative Testing to find Latent Flaws

Validation

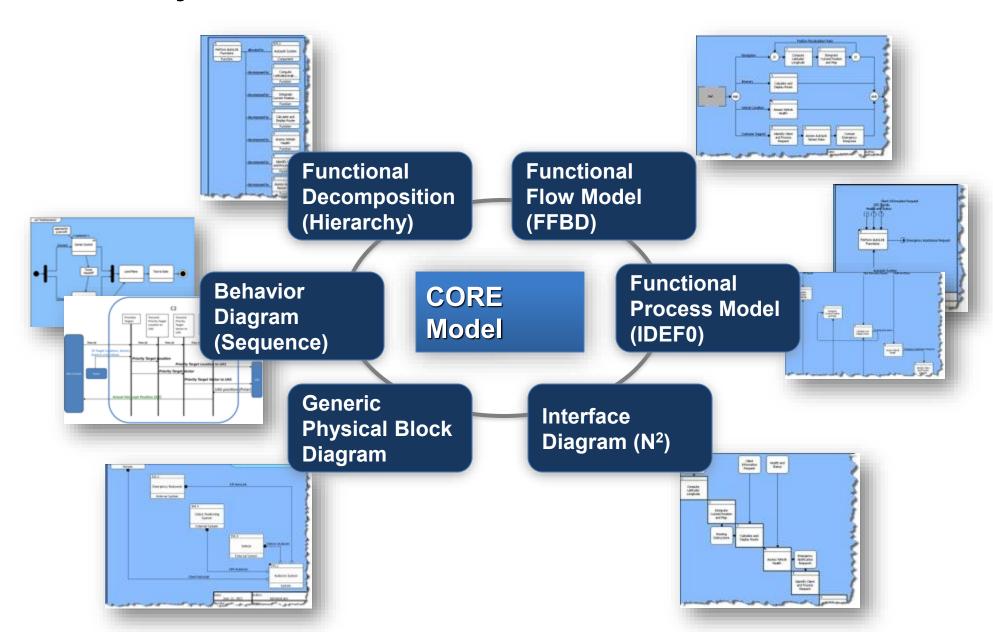
- Validation: "Did I design or build the Right System?"
- Validation shows that the Design when used according to the Operations Concept meets the Requirements and the Customers Goals and Objectives and can be produced within the Cost, Schedule and Risk constraints
- Validation Methods: Analysis, Predictions, Trade Studies, Test
- The requirements flow is also validated to show that "Parent" requirements have valid "Child" requirements, and that "Orphan" requirements are not driving the system design or implementation.
- Initial Validation during Phase A and B is critical to proceeding into Phase C where detail design occurs
 - Otherwise, the detail design proceeds on the "Wrong" system
- Validation also occurs in parallel with verification where End to End Tests, Mission Simulations show that the "Right System" has been built

HSI Defined

Human Systems Integration is defined as a process that optimizes the human part of the total system equation by integrating human factors engineering, manpower, personnel, training, health, safety, survivability, and habitability considerations into the system acquisition process



What is a System Model?



Important IEEE standards

ISO/IEC TR 24748	Systems and software engineering—Life cycle management—Part 1, guide for life cycle management; Part 2, guide to the application
	of ISO/IEC 15288 (system life cycle processes)
ISO/IEC/IEEE 24765	Systems and software engineering—Vocabulary
ISO/IEC/IEEE 29148	Software and systems engineering—Life cycle processes—Requirements engineering
ISO/IEC/IEEE 42010	Systems and software engineering—Architecture description (replaces IEEE 1471)
ISO 10303-233	Industrial automation systems and integration—Product data representation and exchange—

ISO/IEC/IEEE 15288	Systems and software engineering—System life cycle processes
ANSI/EIA-632	Processes for engineering a system
ISO/IEC/IEEE 26702	Systems engineering— Application and management
ISO/IEC/IEEE 15289	Systems and software engineering—Content of systems and software life cycle process information products (documentation)
ISO/IEC/IEEE 15939	Systems and software engineering—Measurement process
ISO/IEC/IEEE 16085	Systems and software engineering—Life cycle processes—Risk management
ISO/IEC/IEEE 16326	Systems and software engineering—Life cycle

Decision gate

It is an approval event in the project cycle, outcomes are:



- Acceptable: Proceed with the next stage of the project.
- Acceptable with reservations: Proceed and respond to action items.
- *Unacceptable: Do not proceed*—continue this stage and repeat the review when ready.
- Unacceptable: Return to a preceding stage.
- Unacceptable: Put a hold on project activity.
- Unsalvageable: Terminate the project.

Decision gate documentation

- Purpose and scope of the decision gate
- Entry and exit criteria
- Host and chairperson
- Attendees
- Location
- Agenda and how the decision gate is to be conducted
- Evidence to be evaluated
- Actions resulting from the decision gate
- Method of closing the review, including timing for resolution of open action items

Taylor, Travis S..
Introduction to Rocket
Science and Engineering.
CRC Press.

Checks Through the Life Cycle

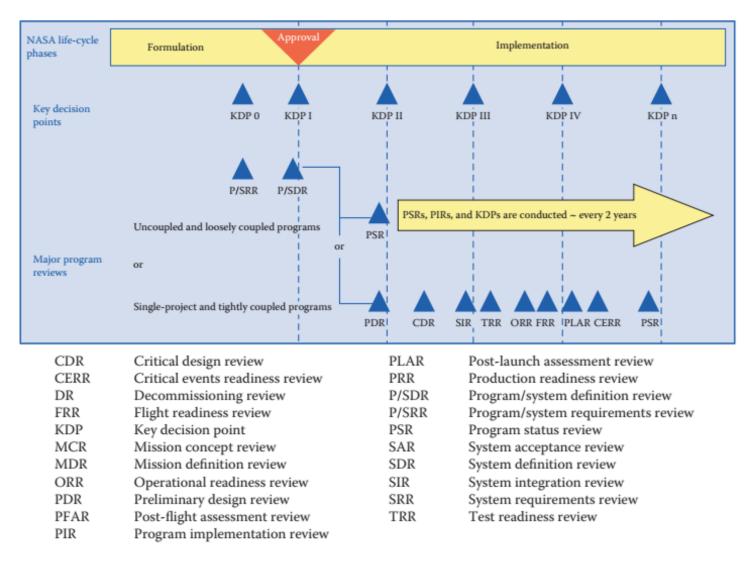


FIGURE 6.1

NASA program life cycle shows the steps of a large-scale development space program. (Courtesy of NASA.)

Reviews

A minimum set of timed reviews should include a systems requirements review (SRR), a preliminary design review (PDR), a critical design review (CDR), a test readiness review (TRR), an operational readiness review (ORR), an operational capability review (OCR), lifecycle assessment reviews (LAR), and a retirement and disposal review (RDR). The reviews need to be set up to include specific program accomplishments, transition decisions, and completed documentation. These reviews occur on a timeline and are embedded in a schedule. Normally, the milestone reviews are mapped to program phases that establish the entire program timeline from conception to birth to retirement (pre-cradle to grave).

• Boord, Warren J.; Hoffman, John B.. Air and Missile Defense Systems Engineering CRC Press.

NIST SP 800-27 – Life Cycle

- Initiation: During the initiation phase, the need for a system is expressed and the purpose of the system is documented. Activities include conducting an impact assessment in accordance with FIPS-199 (http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf).
- **Development/Acquisition:** During this phase, the system is designed, purchased, programmed, developed, or otherwise constructed. This phase often consists of other defined cycles, such as the system development cycle or the acquisition cycle. Activities include determining security requirements, incorporating security requirements into specifications, and obtaining the system.
- Implementation: During implementation, the system is tested and installed or fielded. Activities include installing/turning on controls, security testing, certification, and accreditation.
- **Operation/Maintenance:** During this phase, the system performs its work. Typically, the system is also being modified by the addition of hardware and software and by numerous other events. Activities include security operations and administration, operational assurance, and audits and monitoring.
- **Disposal**: The disposal phase of the IT system life-cycle involves the disposition of information, hardware, and software. Activities include moving, archiving, discarding or destroying information and sanitizing the media.

Enabling Systems

- Enabling systems are systems that facilitate the life cycle activities of the SOI. The enabling systems provide services that are needed by the SOI during one or more life cycle stages, although the enabling systems are not a direct element of the operational environment. Examples of enabling systems include collaboration development systems, production systems, logistics support systems, etc. They enable progress of the SOI in one or more of the life cycle stages.
- -INCOSE. INCOSE Systems Engineering Handbook (p. 10). Wiley.

System of Systems

- A "system of systems" (SoS) is an SOI whose elements are managerially and/ or operationally independent systems. These interoperating and/ or integrated collections of constituent systems usually produce results unachievable by the individual systems alone. Because an SoS is itself a system, the systems engineer may choose whether to address it as either a system or as an SoS, depending on which perspective is better suited to a particular problem.
- INCOSE. INCOSE Systems Engineering Handbook (p. 8). Wiley.

Mission Analysis process

The purpose of the Mission Analysis process is to define the mission problem or opportunity, characterize the solution space, and determine potential solution class(es) that could address a problem or take advantage of an opportunity.

• ISO/IEEE 15288

NASA Systems Engineering Competency Model

TABLE 2.7-1 NASA System Engineering Competency Model

		BLE 2.7-1 NASA System Engineering Competency Model
Competency Area	Competency	Description
SE 1.0 System Design	SE 1.1 Stakeholder Expectation Definition & Management	Eliciting and defining use cases, scenarios, concept of operations and stakeholder expectations. This includes identifying stakeholders, establishing support strategies, establishing a set of Measures of Effectiveness (MOEs), validating stakeholder expectation statements, and obtaining commitments from the customer and other stakeholders, as well as using the baselined stakeholder expectations for product validation during product realization
	SE 1.2 Technical Requirements Definition	Transforming the baseline stakeholder expectations into unique, quantitative, and measurable technical requirements expressed as "shall" statements that can be used for defining the design solution. This includes analyzing the scope of the technical problems to be solved, defining constraints affecting the designs, defining the performance requirements, validating the resulting technical requirement statements, defining the Measures of Performance (MOPs) for each MOE, and defining appropriate Technical Performance Measures (TPMs) by which technical progress will be assessed.
	SE 1.3 Logical Decomposition	Transforming the defined set of technical requirements into a set of logical decomposition models and their associated set of derived technical requirements for lower levels of the system, and for input to the design solution efforts. This includes decomposing and analyzing by function, time, behavior, data flow, object, and other models. It also includes allocating requirements to these decomposition models, resolving conflicts between derived requirements as revealed by the models, defining a system architecture for establishing the levels of allocation, and validating the derived technical requirements.
	SE 1.4 Design Solution Definition	Translating the decomposition models and derived requirements into one or more design solutions, and using the Decision Analysis process to analyze each alternative and for selecting a preferred alternative that will satisfy the technical requirements. A full technical data package is developed describing the selected solution. This includes generating a full design description for the selected solution; developing a set of 'make-to,' 'buy-to,' 'reuse-to,' specifications; and initiating the development or acquisition of system products and enabling products.
SE 2.0 Product Realization	SE 2.1 Product Implementation	Generating a specific product through buying, making, or reusing so as to satisfy the design requirements. This includes preparing the implementation strategy; building or coding the produce; reviewing vendor technical information; inspecting delivered, built, or reused products; and preparing product support documentation for integration.
	SE 2.2 Product Integration	Assembling and integrating lower-level validated end products into the desired end product of the higher-level product. This includes preparing the product integration strategy, performing detailed planning, obtaining products to integrate, confirming that the products are ready for integration, preparing the integration environment, and preparing product support documentation.
	SE 2.3 Product Verification	Proving the end product conforms to its requirements. This includes preparing for the verification efforts, analyzing the outcomes of verification (including identifying anomalies and establishing recommended corrective actions), and preparing a product verification report providing the evidence of product conformance with the applicable requirements.

-NASA Systems Engineering Handbook

NASA Systems Engineering Competency Model

Competency Area	Competency	Description
SE 2.0 Product Realization	SE 2.4 Product Validation	Confirming that a verified end product satisfies the stakeholder expectations for its intended use when placed in its intended environment and ensuring that any anomalies discovered during validation are appropriately resolved prior to product transition. This includes preparing to conduct product validation, performing the product validation, analyzing the results of validation (including identifying anomalies and establishing recommended corrective actions), and preparing a product validation report providing the evidence of product conformance with the stakeholder expectations baseline.
	SE 2.5 Product Transition	Transitioning the verified and validated product to the customer at the next level in the system structure. This includes preparing to conduct product transition, evaluating the product and enabling product readiness for product transition, preparing the product for transition (including handling, storing, and shipping preparation), preparing sites, and generating required documentation to accompany the product
SE 3.0 Technical Management	SE 3.1 Technical Planning	Planning for the application and management of each common technical process, as well as identifying, defining, and planning the technical effort necessary to meet project objectives. This includes preparing or updating a planning strategy for each of the technical processes, and determining deliverable work products from technical efforts; identifying technical reporting requirements; identifying entry and success criteria for technical reviews; identifying product and process measures to be used; identifying critical technical events; defining cross domain interoperability and collaboration needs; defining the data management approach; identifying the technical risks to be addressed in the planning effort; identifying tools and engineering methods to be employed; and defining the approach to acquire and maintain technical expertise needed. This also includes preparing the Systems Engineering Management Plan (SEMP) and other technical plans; obtaining stakeholder commitments to the technical plans; and issuing authorized technical work directives to implement the technical work
	SE 3.2 Requirements Management	Managing the product requirements, including providing bidirectional traceability, and managing changes to establish requirement baselines over the life cycle of the system products. This includes preparing or updating a strategy for requirements management; selecting an appropriate requirements management too; training technical team members in established requirement management procedures; conducting expectation and requirements traceability audits; managing expectation and requirement change information
	SE 3.3 Interface Management	Establishing and using formal interface management to maintain internal and external interface definition and compliance among the end products and enabling products. This includes preparing interface management procedures, identifying interfaces, generating and maintaining interface documentation, managing changes to interfaces, disseminating interface information, and conducting interface control
	SE 3.4 Technical Risk Management	Examining on a continual basis the risks of technical deviations from the plans, and identifying potential technical problems before they occur. Planning, invoking, and performing risk-handling activities as needed across the life of the product or project to mitigate impacts on meeting technical objectives. This includes developing the strategy for technical risk management, identifying technical risks, and conducting technical risk assessment; preparing for technical risk mitigation, monitoring the status of each technical risk, and implementing technical risk mitigation and contingency action plans when applicable thresholds have been triggered.

NASA Systems Engineering Competency Model

Competency Area	Competency	Description
SE 3.0 Technical Management	SE 3.5 Configuration Management	Identifying the configuration of the product at various points in time, systematically controlling changes to the configuration of the product, maintaining the integrity and traceability of product configuration, and preserving the records of the product configuration throughout its life cycle. This includes establishing configuration management strategies and policies, identifying baselines to be under configuration control, maintaining the status of configuration documentation, and conducting configuration audits
	SE 3.6 Technical Data Management	Identifying and controlling product-related data throughout its life cycle; acquiring, accessing, and distributing data needed to develop, manage, operate, support, and retire system products; managing and disposing data as records; analyzing data use; obtaining technical data feedback for managing the contracted technical efforts; assessing the collection of appropriate technical data and information; maintaining the integrity and security of the technical data, effectively managing authoritative data that defines, describes, analyzes, and characterizes a product life cycle; and ensuring consistent, repeatable use of effective Product Data and Life-cycle Management processes, best practices, interoperability approaches, methodologies, and traceability. This includes establishing technical data management strategies and policies; maintaining revision, status, and history of stored technical data and associated metadata; providing approved, published technical data; providing technical data to authorized parties; and collecting and storing required technical data.
	SE 3.7 Technical Assessment	Monitoring progress of the technical effort and providing status information for support of the system design, product realization, and technical management efforts. This includes developing technical assessment strategies and policies, assessing technical work productivity, assessing product quality, tracking and trending technical metrics, and conducting technical, peer, and life cycle reviews.
	SE 3.8 Technical Decision Analysis	Evaluating technical decision issues, identifying decision criteria, identifying alternatives, analyzing alternatives, and selecting alternatives. Performed throughout the system life cycle to formulate candidate decision alternatives, and evaluate their impacts on health and safety, technical, cost, and schedule performance. This includes establishing guidelines for determining which technical issues are subject to formal analysis processes; defining the criteria for evaluating alternative solutions; identifying alternative solutions to address decision issues; selecting evaluation methods; selecting recommended solutions; and reporting the results and findings with recommendations, impacts, and corrective actions.

DoD Goals of Digital Engineering

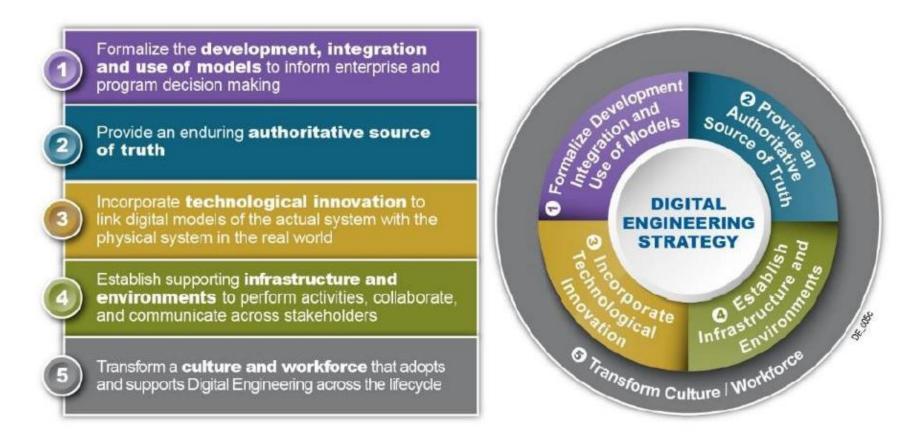
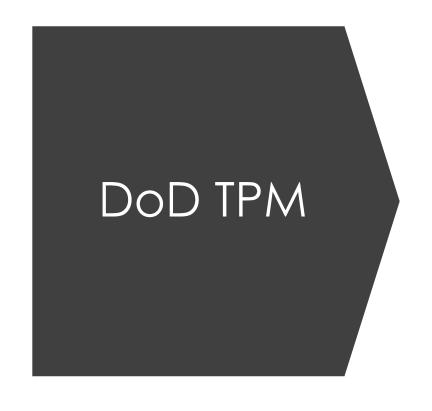


Figure 2-1. Five Goals of DoD's Digital Engineering Strategy



Core Technical Performance Measure (TPM) Category	Description of TPM
Schedule Management	Include metrics to assess both schedule health (e.g., the Defense Contract Management Agency 14-point health check), associated completeness of the Work Breakdown Structure and the risk register. A healthy, complete and risk-enabled schedule forms the technical basis for the Earned Value Management System (EVMS). Strong schedule metrics are paramount for accurate EVMS data.
Staffing and Personnel Management	Metrics should evaluate the adequacy of the effort, skills, experience and quantity of personnel assigned to the program to meet management objectives throughout the acquisition life cycle.
Resource Management	Metrics should evaluate the adequacy of resources and/or tools (e.g., models, simulations, automated tools, synthetic environments) to support the schedule. See also Table 5-7: Product Support Considerations.
Software Development Management	Metrics should evaluate software development progress against the software development plan. For example, the rate of code generation (lines of code per man-hour). (See Section 2.2.4 Software Engineering)
Software Quality	Metrics should address software technical performance and quality (e.g., defects, rework) evaluating the software's ability to meet user needs. (See Section 2.2.4 Software Engineering)
Requirements Management	Evaluate the stability and adequacy of the requirements to provide the required capability, on-time and within budget. Includes the growth, change, completeness and correctness of system requirements. (See Section 4.1.4 Requirements Management Process)
Risk Management	Metrics should include the number of risks open over time or an aggregate of risk exposure (the potential impact to the performance, cost and schedule). (See Section 4.1.5 Risk Management Process)
Test Management	Metrics should include measures of the stability of the verification and validation process (e.g., number of test points, development of test vignettes and test readiness).

-DoD Systems Engineering Guidebook

DoD TPM Hierarchy

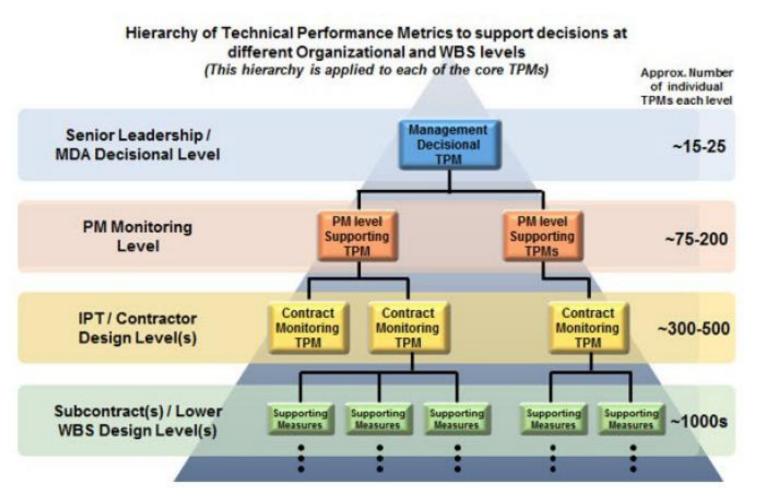


Figure 4-5. TPM Hierarchy

Naval Integrated Modeling Environment

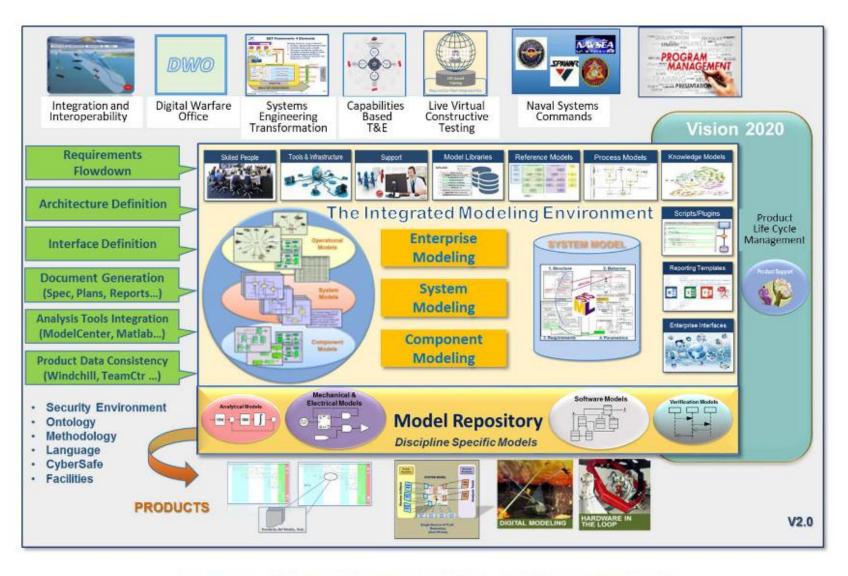


Figure 1 - Naval Enterprise Integrated Modeling Environment



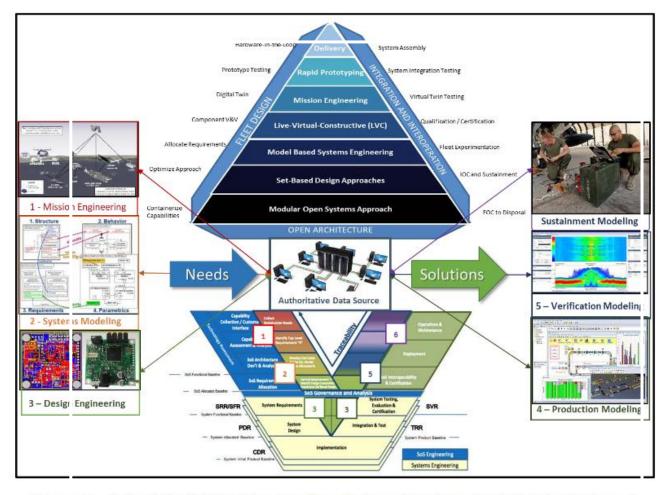


Figure 3 – Role of Digital Models connecting System of Systems Analysis, Standards, and Model Based Systems Engineering

Technical Readiness

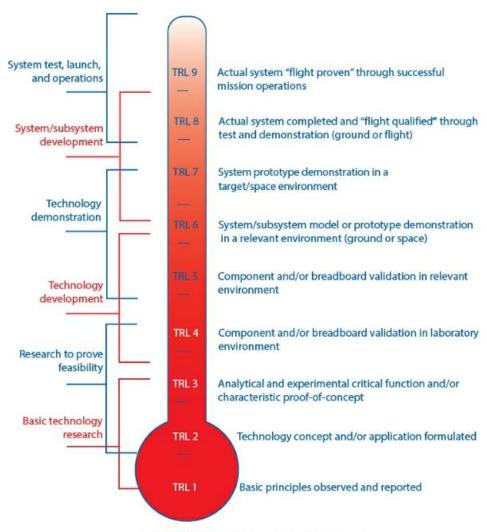


FIGURE G.4-1 Technology Readiness Levels

-DoD Systems Engineering Guidebook

DoD TPM Hierarchy

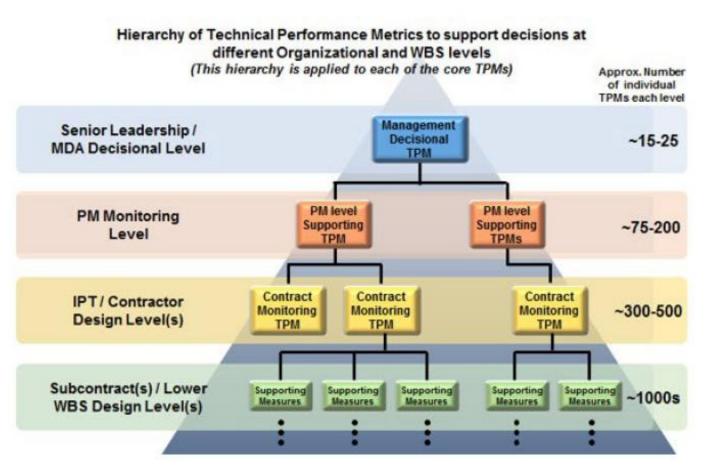


Figure 4-5. TPM Hierarchy

Systems Engineering Problem SolvingOntology

- In information science, ontology refers to a formal, explicit specification of a shared conceptualization of a domain. Put simply, it is a structured framework that defines the entities, relationships, and rules relevant to a particular area of knowledge.
- While the term has roots in philosophy (study of "being" and categories of existence), in information science it has a more practical and technical meaning.

Systems Engineering Problem Solving -Ontology

Core Elements of an Ontology

- **1.Concepts (Classes/Entities):** The categories or types of things in a domain.
 - 1. Example: Person, Document, Organization.
- **2.Relationships (Properties):** How concepts are connected to one another.
 - 1. Example: "Person writes Document", "Organization employs Person".
- **3.Attributes (Data Properties):** Characteristics or properties of concepts.
 - 1. Example: A *Person* has a *Name* or *Birthdate*.
- **4.Instances (Individuals):** Specific examples of concepts.
 - 1. Example: "Marie Curie" as an instance of Person.

Zwicky Box

• A Zwicky Box—named after the Swiss astrophysicist Fritz Zwicky—is essentially another name for morphological analysis, and it's very similar in spirit to the Hall Morphological Box, but it comes from a slightly different tradition and is used even more explicitly for problem-solving and innovation. It is a a multi-dimensional matrix that organizes the parameters of a complex problem and lists all possible states for each parameter, so you can systematically explore every combination. It's designed to help uncover non-obvious solutions by breaking down a problem into independent dimensions and recombining their options.



Just like the Hall Morphological Box, a Zwicky Box:

Zwicky Box

Has parameters (factors, attributes, or dimensions) along one axis.

Has possible values or variants for each parameter listed below it.

Any single complete combination of values—one from each parameter—represents a potential solution or system configuration.

Zwicky Box

• The following is an example for a drone

Parameter	Option 1	Option 2	Option 3	Option 4
Power Source	Battery	Fuel Cell	Hybrid	Solar
Propulsion Type	Fixed-wing	Quadcopter	Hexacopter	VTOL
Frame Material	Carbon Fiber	Aluminum	Plastic	Composite
Payload Capacity	<1 kg	1-5 kg	5-20 kg	20+ kg
Control System	Manual	Semi-autonomous	Fully autonomous	
Range	<5 km	5–20 km	20-50 km	50+ km

Hall Morphological Box

- The Hall Morphological Box in systems engineering is a conceptual framework developed by Arthur D. Hall (one of the pioneers of systems engineering) to help structure and explore complex problems systematically.
- It's essentially a matrix-based method—a type of "morphological analysis"—that organizes all possible combinations of a system's parameters and attributes. A Hall Morphological Box typically looks like a multi-dimensional table (or hypercube), where:
 - Columns (or axes) represent parameters or features of the system.
 - Rows under each column represent possible states, options, or values for that parameter.
- By selecting one value from each column, you define a possible system configuration.

Hall Morphological Box

• The following is an example for a drone

Parameter	Option 1	Option 2	Option 3
Power Source	Battery	Fuel Cell	Solar
Propulsion Type	Fixed-wing	Rotor	Hybrid
Control System	Manual	Semi-auto	Fully-auto

Zwicky VS Hall Box

Zwicky's approach is problem-solving and creative exploration—often used for totally novel solution spaces.

Hall's approach is more structured for engineering trade studies.

Zwicky emphasized eliminating infeasible combinations early to reduce complexity.

Hall's method often keeps more possibilities open until more detailed evaluation.

Battlespace analysis

To complete a preliminary design level battlespace analysis, we first consider first defining battlespace depth of fire (DOF) or firepower followed by an engagement analysis. Defining the DOF requires determining for each AMD preliminary design configuration where, how many, and which interceptor variants can reach the target sets. The engagement analysis will tell us which interceptors and variants can successfully engage the targets and how many it will take to achieve the system Pk requirement. The target set is defined by speed, altitude, signature, and other environmental considerations for the battlespace evaluation. The engagement analysis requires the addition of any target defense penetration features that are uniquely intended to defeat the interceptor such as evasive maneuver.

Boord, Warren J.; Hoffman, John B.. Air and Missile Defense Systems Engineering. CRC Press.



0 Capabilities Based Assessment (CBA)

 A Joint Capabilities Integration and Development System (JCIDS) analytic process. The CBA identifies capability requirements and associated capability gaps. Results of a CBA or other study provide the source material for one or more Initial Capabilities Documents (ICDs), or other JCIDS documents in certain cases when an ICD is not required

 A number of DoDAF views are to be used to capture results of a CBA, facilitating reuse in JCIDS documents, acquisition activities, and capability portfolio management. When one or more studies or analyses are used in place of a CBA, the Sponsor may need to consolidate the data from those studies into a single set of DoDAF products appropriate for the scope of the ICD.

Common Logic

- Common Logic (CL) is an ISO/IEC international standard (ISO/IEC 24707:2007) that specifies a framework for a family of logic-based languages designed for knowledge representation and exchange.
- Its main goal is to allow different logical systems and notations to interoperate, meaning that knowledge expressed in one system can be translated into another without losing meaning.

Common Logic

CLIF (Common Logic Interchange Format)

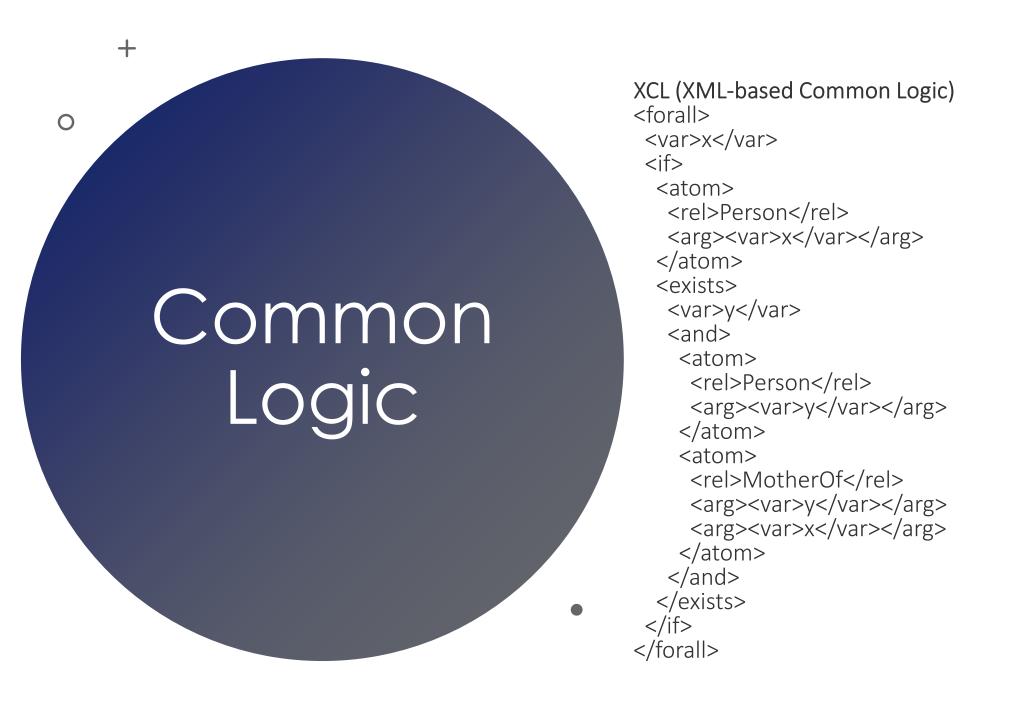
+

0

Common Logic

CGIF (Conceptual Graph Interchange Format)

```
[Person: *x]  // There exists a person x
-> (MotherOf ?y *x)  // Some y is mother of x
[Person: ?y]  // y is also a person
```



Common Logic

Dialect	Style	Best For	
CLIF	Lisp-like, compact, human-readable for logicians	General logical expression, interchange among researchers	
CGIF	Graph-based, conceptual graph notation	Visual thinkers, conceptual modeling	
XCL	XML-structured, verbose	Machine processing, integration with web/enterprise systems	

Wicked Problem

• Yes, this actually is a real technical term. A wicked problem is a type of problem that is complex, ill-defined, and resistant to resolution. Unlike "tame problems" (which have clear solutions and endpoints). It can refer to a problem that is difficult or impossible to solve because of incomplete, contradictory, and changing requirements that are often difficult to recognize. The term was introduced by Horst Rittel and Melvin Webber (1973) in their paper on social planning, highlighting the challenges of tackling societal and policy issues.