An Introduction to Systems Engineering

Tutorial, INCOSE Brazil Emerging Chapter Event:
Systems Engineering Week
12 December 2011, São José dos Campos, Brasil

and First Half-Day

Systems Engineering for Technology-Based Projects and Product Developments 5-Day Course

Robert J Halligan, FIE Aust

Managing Director, Project Performance International (PPI)
Past Director, International Council on Systems Engineering (INCOSE)
Past INCOSE Head of Delegation to ISO/IEC SC7 on Software and Systems Engineering
Past President, Systems Engineering Society of Australia
Content contributor to EIA/IS-632, EIA 632, IEEE 1220, ISO/IEC 15288 SE standards
Consultant/trainer to BAe Systems, Mitsubishi, EADS, Thales, Raytheon, General Electric and many other enterprises on six continents
Tel: 12 3212 2017 Fax: 12 3212 5582 Cell Phone: 12 8161 7285 email: rhalligan@ppi-int.com

Robert Halligan, FIE Aust

- Past Director, International Council on Systems Engineering (INCOSE)
- Past INCOSE Head of Delegation to ISO/IEC SC7 on Software and Systems Engineering
- Past President, Systems Engineering Society of Australia
- Content contributor to EIA/IS-632, EIA 632, IEEE 1220, ISO/IEC 15288 SE standards
- Consultant/trainer to Alcatel-Lucent, General Dynamics, Mitsubishi, EADS, Thales, Raytheon, General Electric, NATS, IAEA, JAXA, Boeing, and many other enterprises on six continents
- Current INCOSE Ambassador

A SYSTEM IS?

"A regularly interacting or interdependent group of items forming a unified whole"

Source: The Merriam-Webster Unabridged Dictionary

Key concepts:

- Interaction Elements
- Interrelationships
-
- Whole

AN ENGINEERED SYSTEM IS?

"A combination of interacting elements organized to achieve one or more stated purposes"

Source: ISO/IEC 15288:2008 Systems and software engineering — System life cycle processes

Key concepts:

- Organization
- Elements
- Interaction
- Purpose

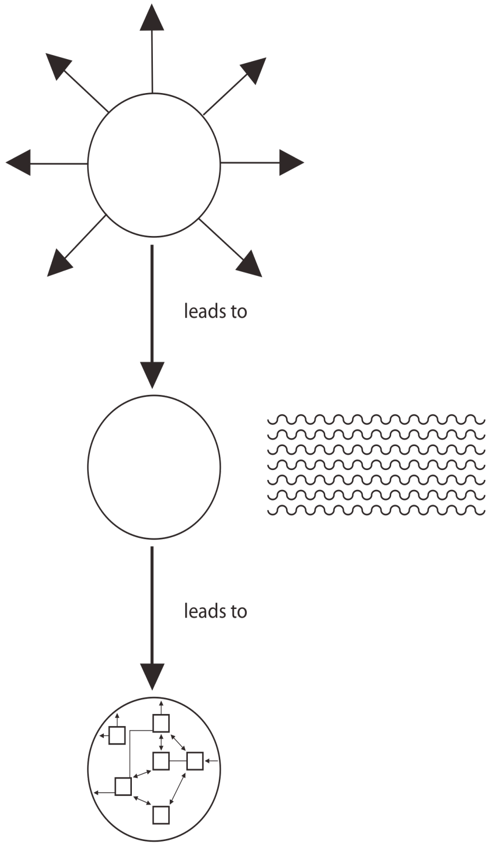
A SYSTEMS APPROACH

System Views

Outward Looking View
(part of a bigger system)

Object View
(object with required and desired characteristics)

Inward Looking View
(seeing the system as a set of interacting objects, the properties of the whole coming from the objects and their interactions)



VIEWS RELATED TO SYSTEMS ENGINEERING

The outward looking view, seeing our system as a part of one or more bigger systems

The object view, seeing our system as an object with a required and desired set of characteristics

The inward looking view, seeing our system as a set of interacting elements

SYSTEMS ENGINEERING IS:

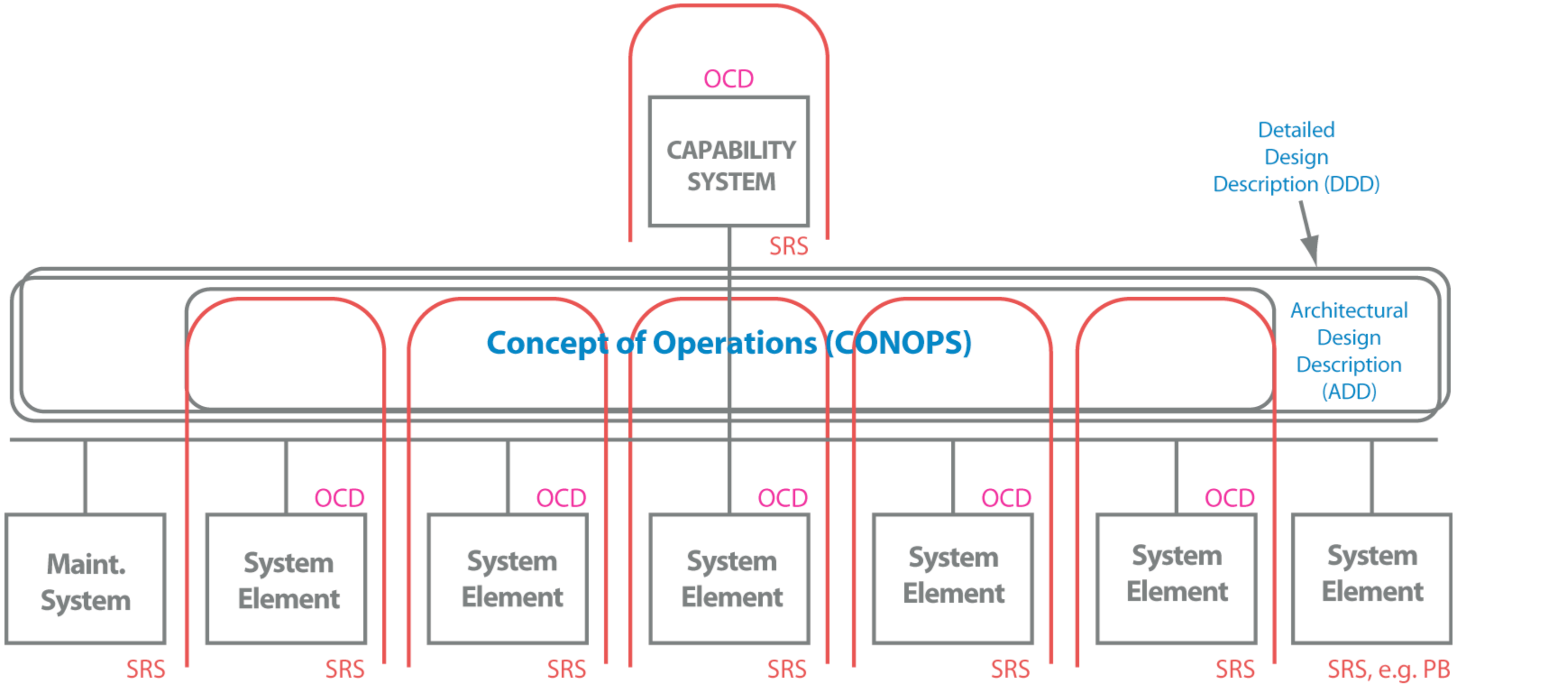
"Systems engineering is an interdisciplinary, collaborative approach to the engineering of systems (of any type) which aims to capture stakeholder needs and objectives and to transform these into a description of a holistic, life-cycle balanced system solution which both satisfies the minimum requirements, and optimizes overall project and system effectiveness according to the values of the stakeholders. Systems engineering incorporates both technical and management processes"

Source: Halligan, 2003

Key concepts:

- Engineering
- Collaboration
- Effectiveness
- Stakeholder satisfaction
- Interdisciplinary working
- Life-cycle balance
- Optimization
- Stakeholder value

PHYSICAL LEVELS RELATED TO KEY INFORMATION TYPES



ADD: Architectural Design Description. An ADD describes the concept of the solution to meet ALL of the requirements of the TTDIS.

CONOPS: Concept of Operations. A CONOPS describes the concept of the solution to meet the subset of the requirements of the TTDIS that are are directly use-related.

DDD: Detailed Design Description. A DDD describes the design to meet ALL of the requirements of the TTDIS. The description is at a level of detail that is implementable, e.g. sufficient to contract for, and/or design and develop, or otherwise acquire, each element of solution at the physical level shown. The DDD incorporates the set of SRSs for the set of system elements, together with instructions for configuration of the set of elements into a whole solution.

OCD: Operational Concept Description. An OCD is a system (subsystem, etc)-centric description of the users of the system, the intended uses of that system, how it is intended the system be used, and the external conditions during which the system will be used. The OCD describes the context within which the problem definition (requirements, MOEs, goals and value relationships) exists.

VRS: Verification Requirements Specification. Specification of the qualities of evidence required that each requirement has been satisfied.

SRS: System Requirements Specification or Software Requirements Specification. A SRS specifies the required characteristics of the item, together with goals (if any) for that item.

PB: Project Brief. The Project Brief is the SRS for the Project System (here shown as Project Infrastructure).

KEY QUESTIONS!

- What should we do in practicing systems engineering?
- What should we not do in practicing systems engineering!?

INDICATORS OF EFFECTIVE SE – PRODUCT-ORIENTED ENTERPRISE:

- On, under or close to development budget
- On, ahead of or close to development schedule
- High Return on Sales
- Market leadership
- Low warranty costs
- Repeat business is the norm
- High staff satisfaction and retention

INDICATORS OF EFFECTIVE SE – CONTRACT-ORIENTED ENTERPRISE:

- On, under or close to development budget
- On, ahead of or close to development schedule
- High contract gross margin
- High customer satisfaction
- Low warranty costs
- Repeat business is the norm
- High staff satisfaction and retention

INDICATORS OF EFFECTIVE SE – INTERNAL PROJECTS:

- On, under or close to development budget
- On, ahead of or close to development schedule
- High internal customer satisfaction
- No desire to outsource
- High staff satisfaction and retention

INDICATORS OF EFFECTIVE SYSTEMS ENGINEERING MANAGEMENT:

- Effective systems engineering
- Harnessing of creativity
- A learning environment
- Growing intellectual capital within the enterprise
- High staff satisfaction and retention
- A shared vision of the product and related focus on quality, cost, time

INDICATORS OF NO SE OR INEFFECTIVE SE:

- Milestones missed
- Significant dispute with customers over requirements
- Many problems and delays occur during system integration
- Significant dispute with customers over testing
- Significant problems occur in released or fielded systems/products
- Engineering effort tends to be back-end loaded during development

PITFALLS IN EXPLOITING SE:

- United States DoD mindset and standards
- “silver bullet” mentality
- Choosing inappropriate resources (e.g. standards, handbooks)
- Forcing new processes on unwilling participants
- Revolution rather than evolution
- Only superficial training of engineering personnel
- No measurement, no IV&V

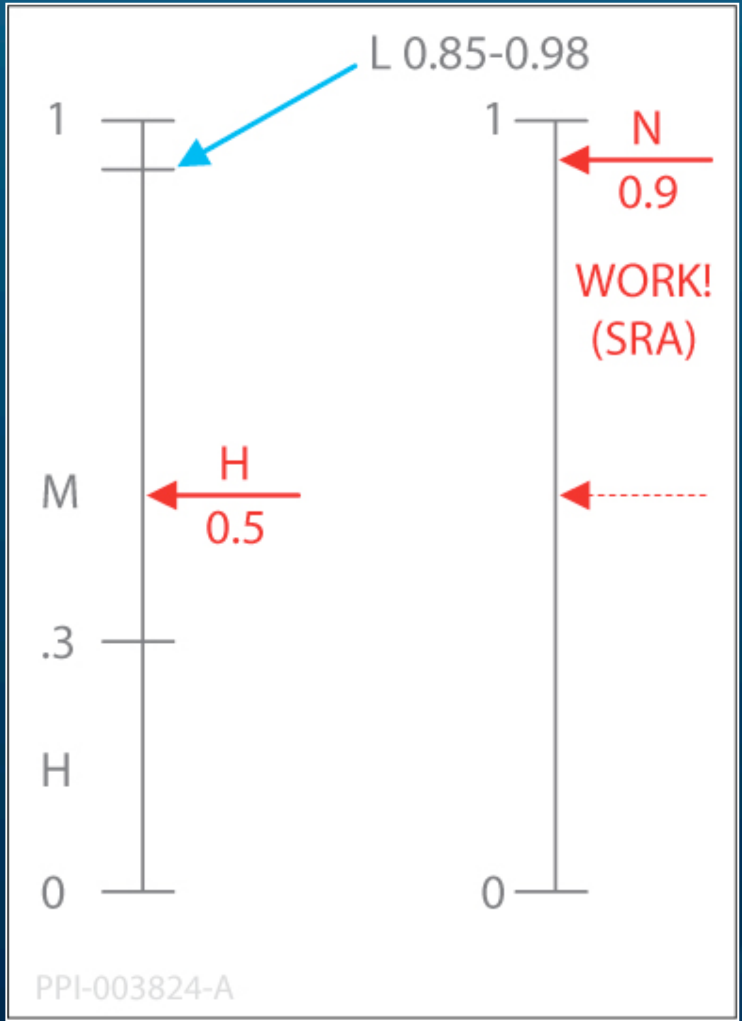
Do:

- Establish an objectively adequate problem definition before committing significant resources to design and development

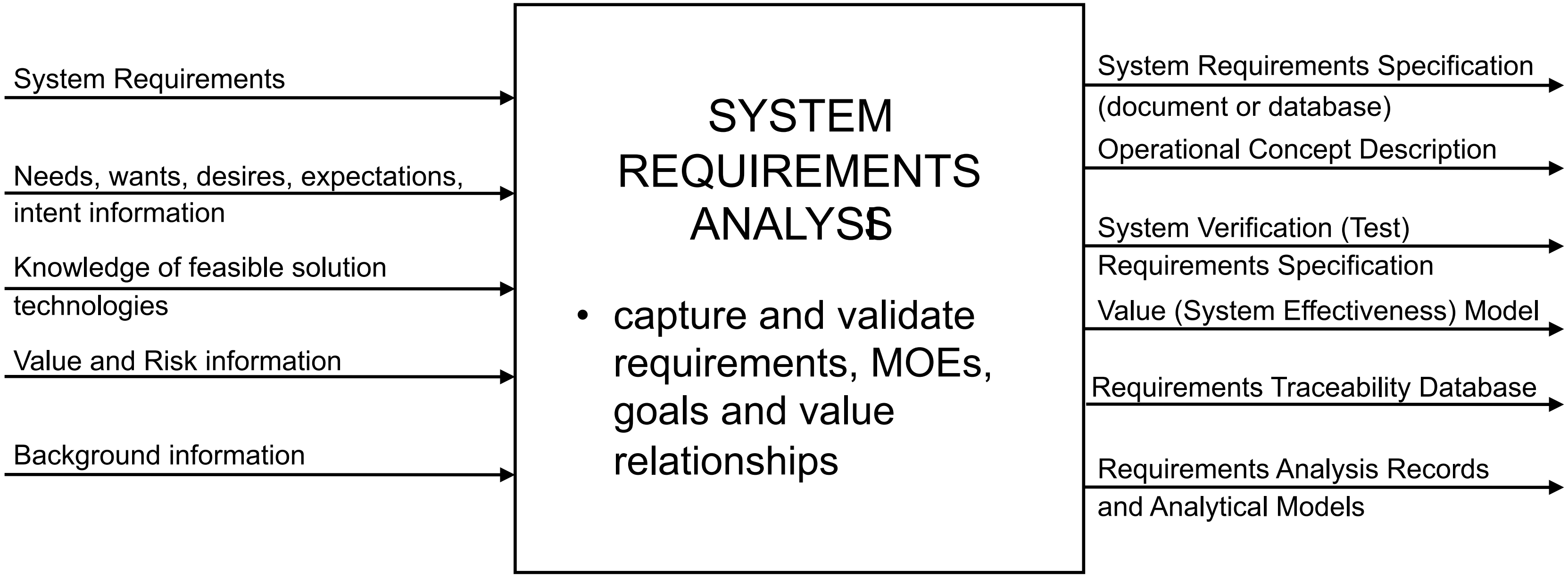
Why?

- Inadequate requirements have consistently been the single biggest cause of project failures and losses in all sectors. The cost of inaction typically exceeds the cost of prevention by a factor of 10-100 to one

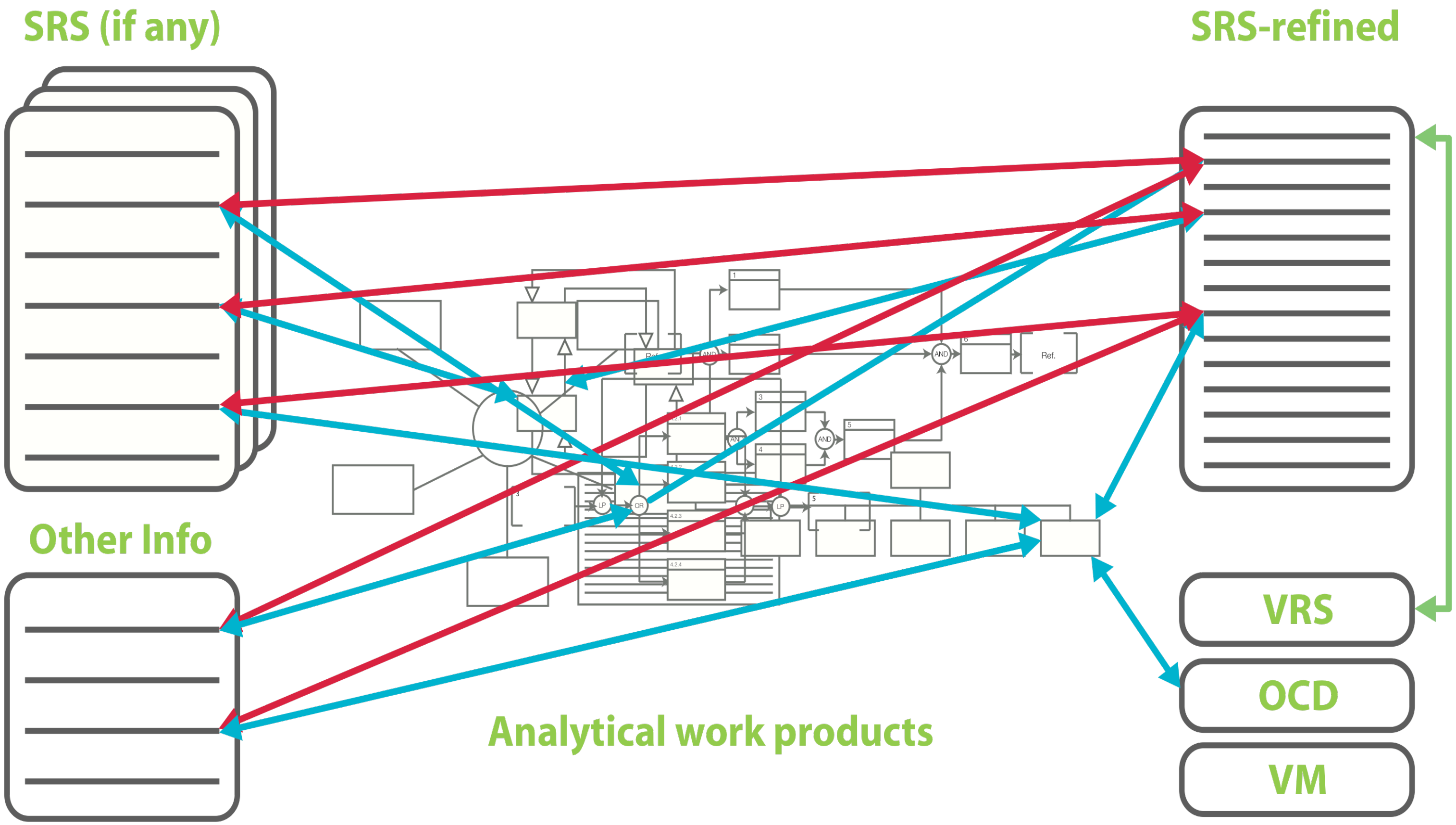
OBJECTIVE CRITERIA FOR ADEQUACY APPLIED TO REQUIREMENTS



SYSTEM REQUIREMENTS ANALYSIS

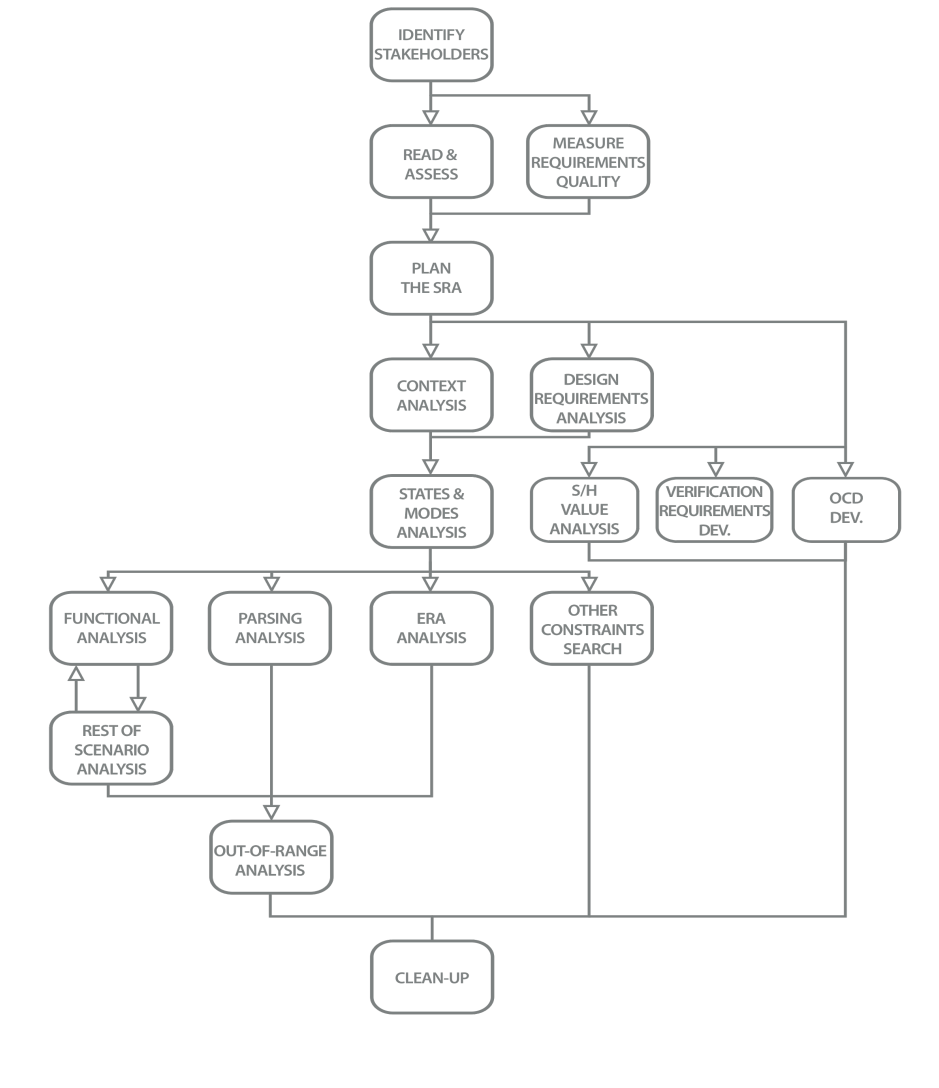


USE ANALYSIS TOGETHER WITH RESOLUTION OF SPECIFIC ISSUES AS THE PRIMARY RA STRATEGY

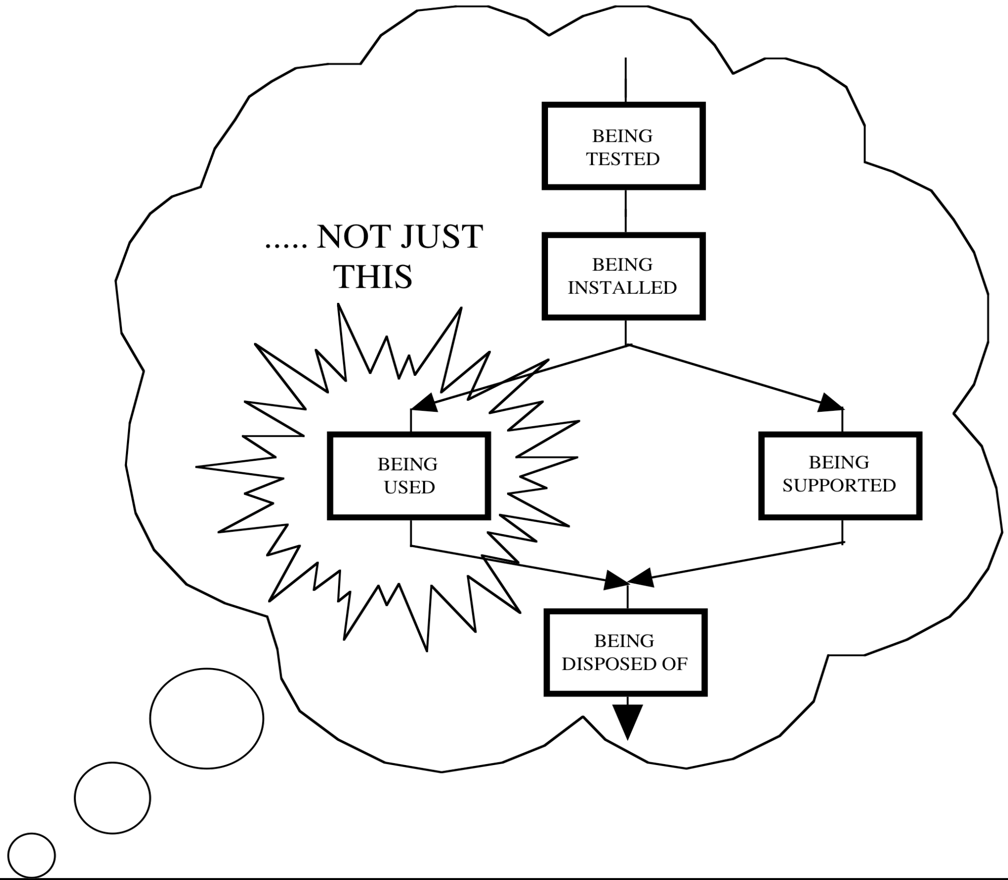


SRS: system or software requirements specification
VRS: verification requirements specification
OCD: operational concept description
VM: value (or system/software effectiveness model)

USE SOUND, EFFECTIVE METHODS OF REQUIREMENTS ANALYSIS



CAPTURE AND VALIDATE REQUIREMENTS ON A LIFECYCLE BASIS

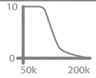
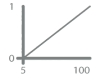
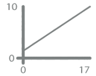
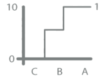
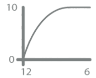
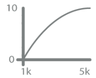
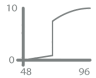
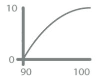
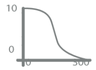


LIFE CYCLE CONTEXT IS IMPORTANT….

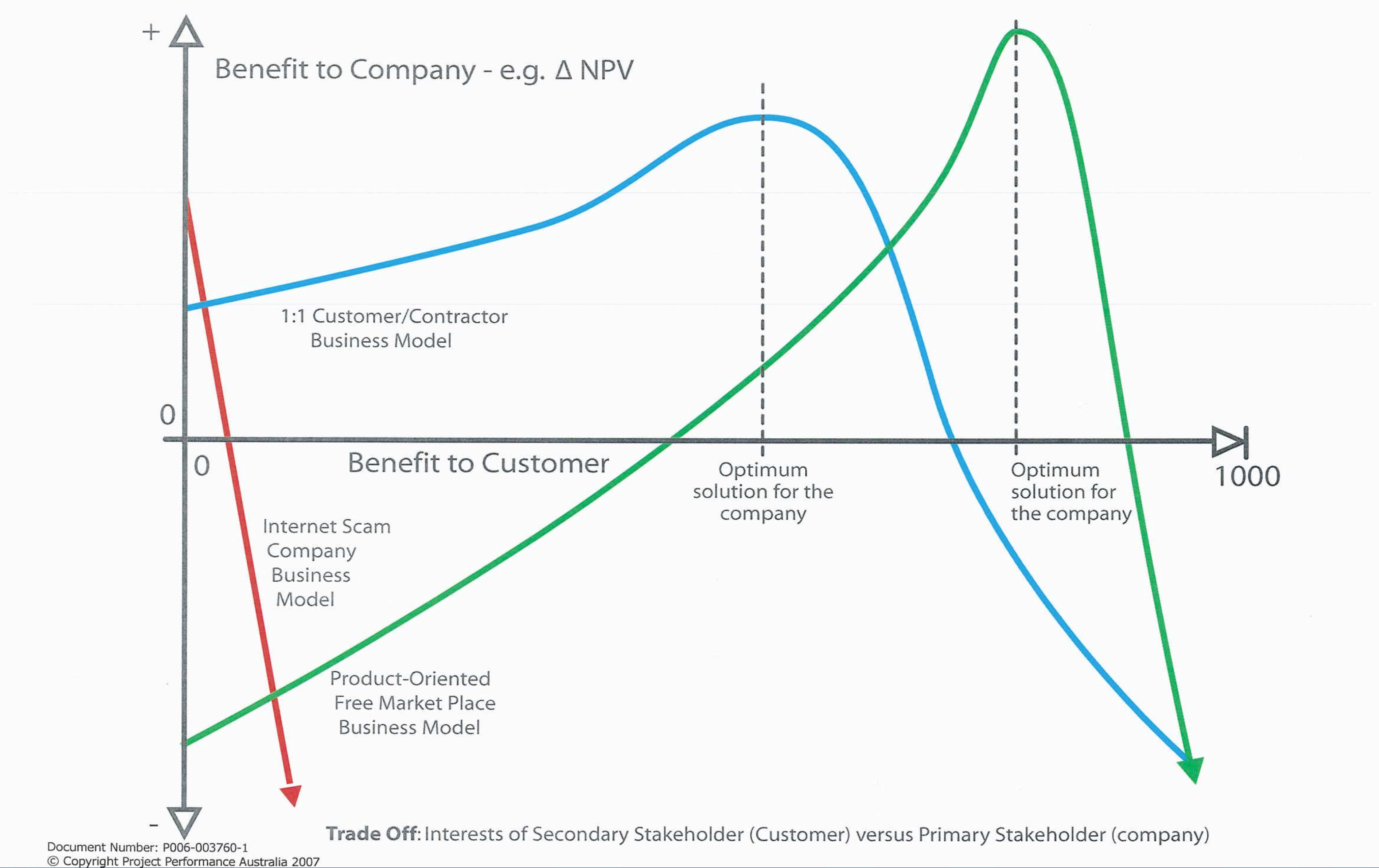
Requirements come from these contexts

CONSIDER MOES & GOALS, NOT JUST REQUIREMENTS

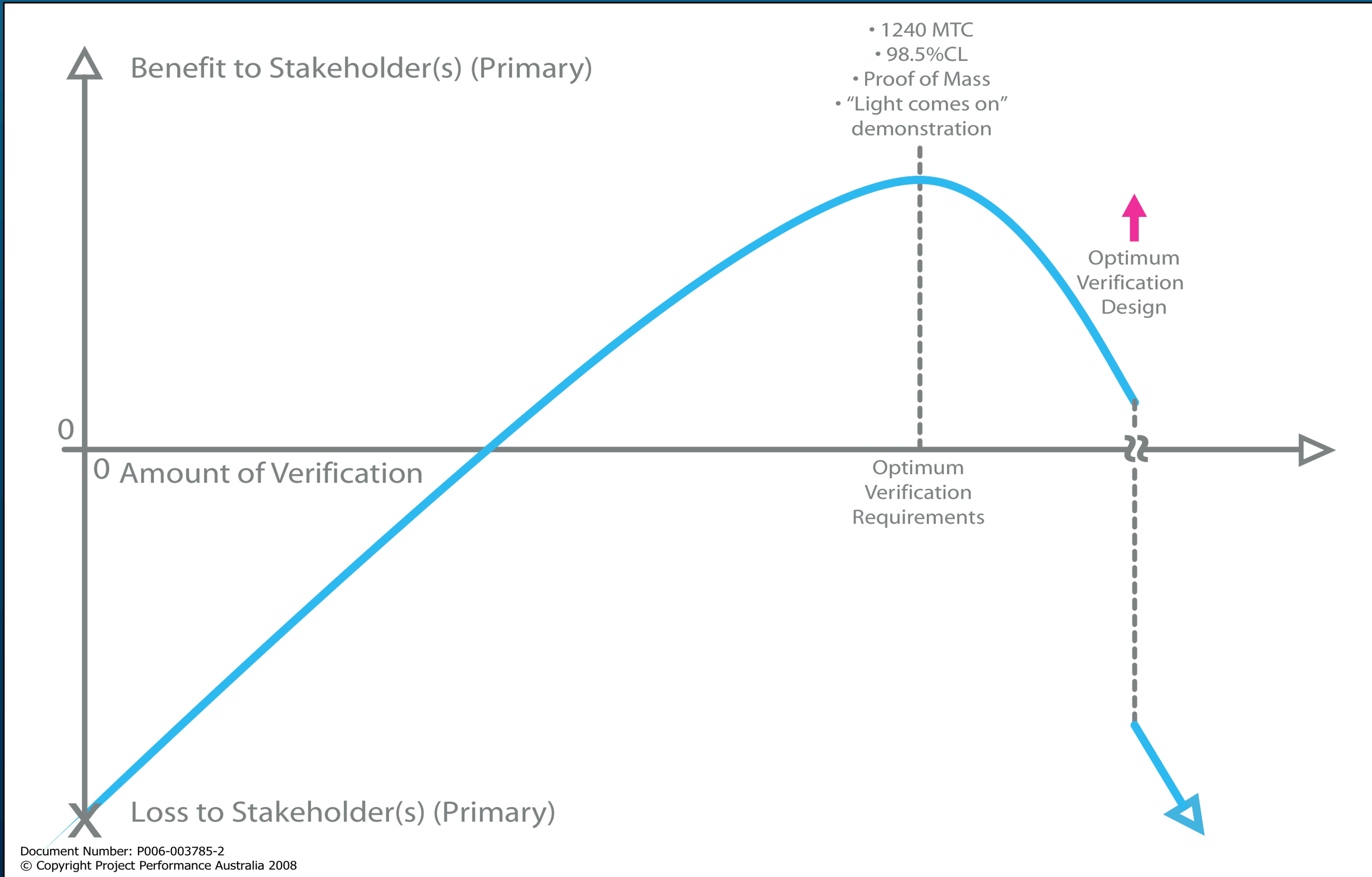
Value (System Effectiveness) Model

MOEs	Worst	Best	Pri	Pts	Weight %	UF
Cost, $k's per unit	200	50	1	100	25	
Reliability, %	95	100	1	100	25	
Interoperability	0	17	7	14	4	
Size(A/B/C)	C	A	8	3	1	
Schedule (MonthS)	12	6	3	40	10	
Visible Optical Range	1000	5000	5	30	7	
Duration of Transmission, hr	48	96	6	27	6	
Readiness, %	90	100	4	39	10	
OS & D Cost, $k pu/10 years	300	10	2	50	12	
				403	100	

MAXIMISE VALUE TO COMPANY, OPTIMISE VALUE TO CUSTOMER



PERFORM OPTIMUM VERIFICATION



Do:

- Design a solution by dividing the big problem into a set of individually sufficiently-well-defined smaller problems, i.e., by defining the required characteristics of each element of the solution (including both product (hardware and software, etc.) and process elements, as applicable)

Why?

- To do otherwise works outside the cognitive limits of the human brain, resulting in design errors, the need for much increased design verification, and problems first revealed in system integration (or later)

Do:

- Apply design skills and technology knowledge in *creating* requirements

Why?

- Every requirement is a part of a solution to a bigger problem, and in a contract context, two bigger problems

Don't:

•*Just* partition requirements, allocating system requirements to system elements, or to categories of technology, e.g. to hardware and software

Why?

- Partitioning system requirements that cannot be implemented by a single system element leads to major problems in system integration, leading, in turn, to cost and schedule blowouts

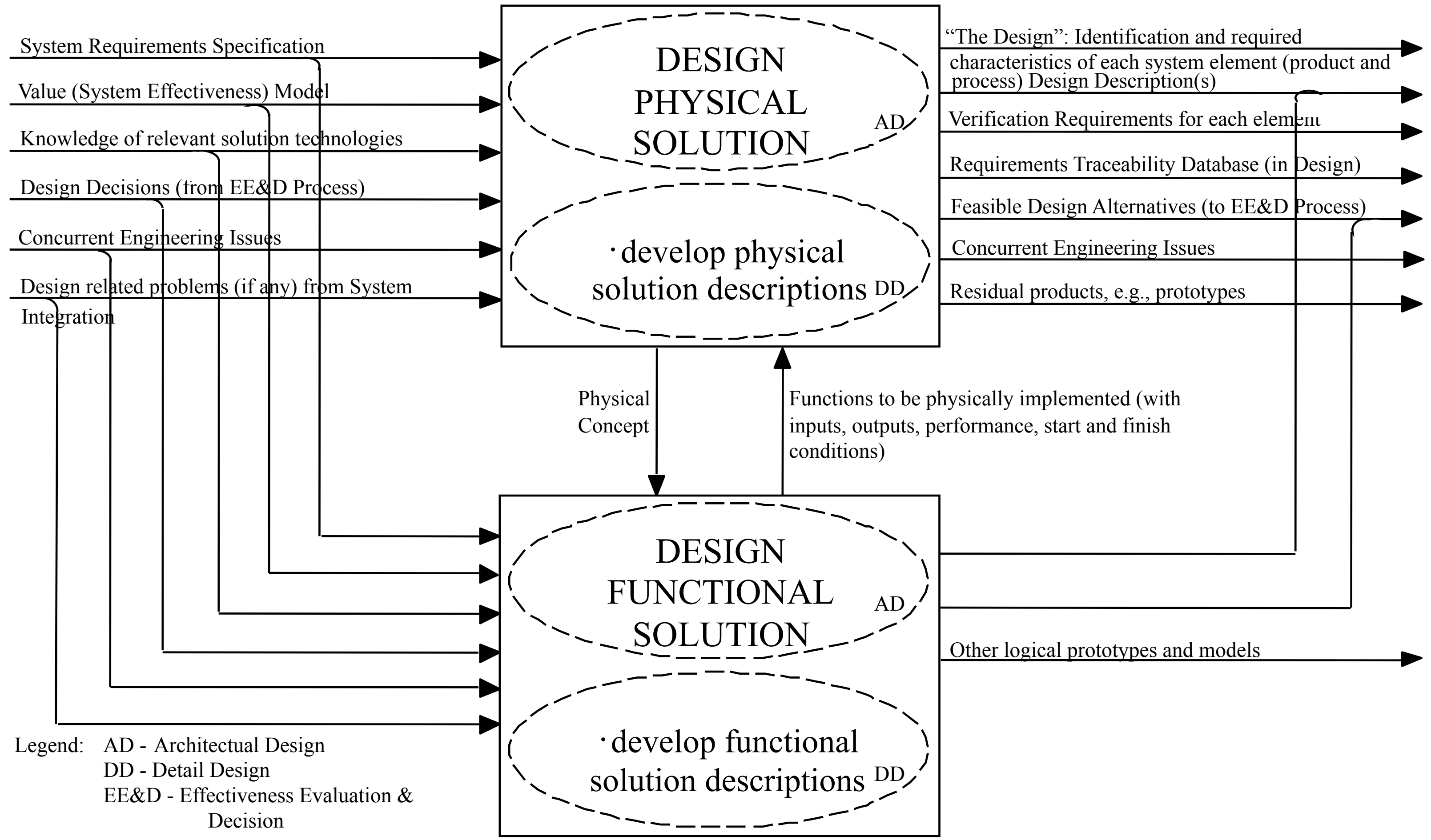
Do:

- Regard knowledge of relevant technologies, and the creativity and attitudes to apply that knowledge in a team environment, as indispensible ingredients of effective systems engineering

Why?

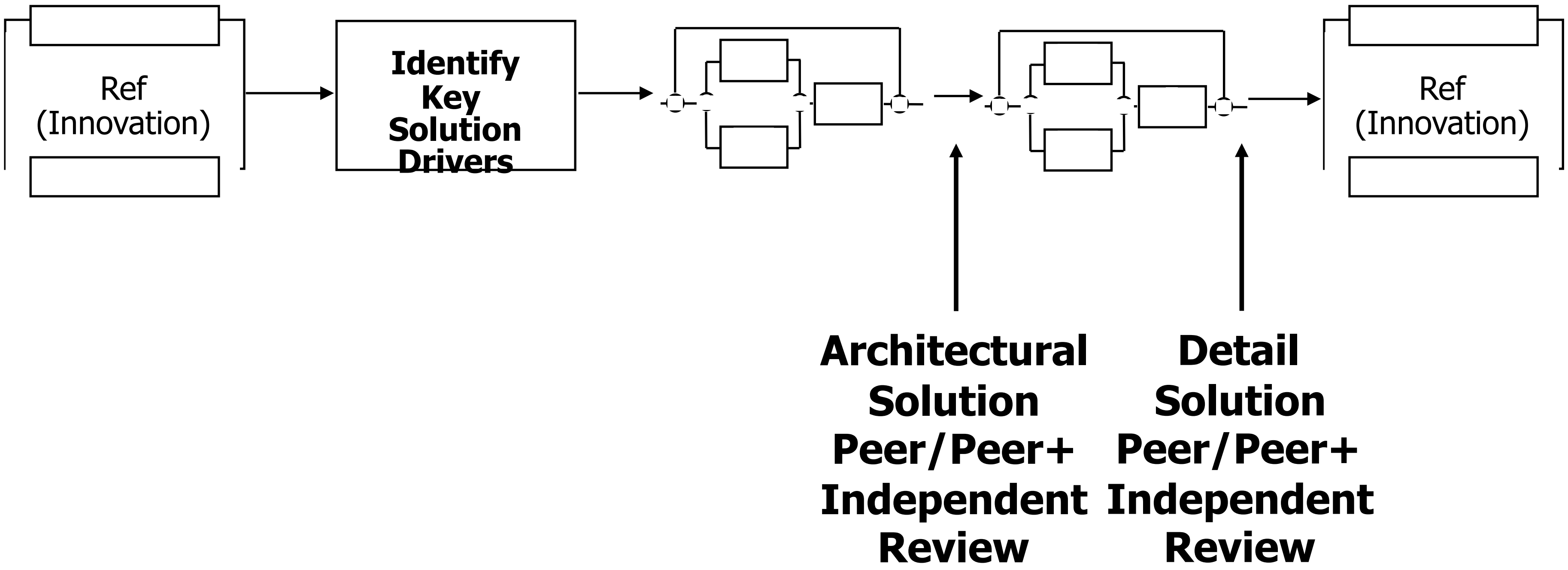
- Process alone is valueless without the knowledge and creativity. Sound processes, selected to match the job at hand, *assist* in transforming good ideas into good solutions, great ideas into great solutions

DESIGN PHYSICAL SOLUTION, DESIGN LOGICAL SOLUTION



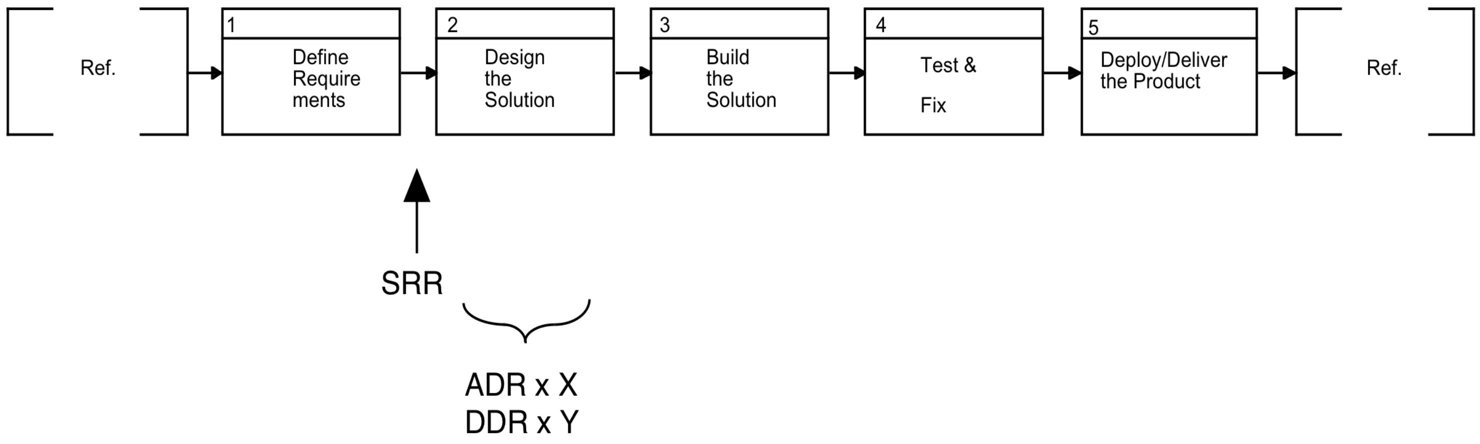
P1135-004875-1

CREATING SOLUTION



Apply a systems approach, always with reference to an object and a physical level of solution

WATERFALL BASICS



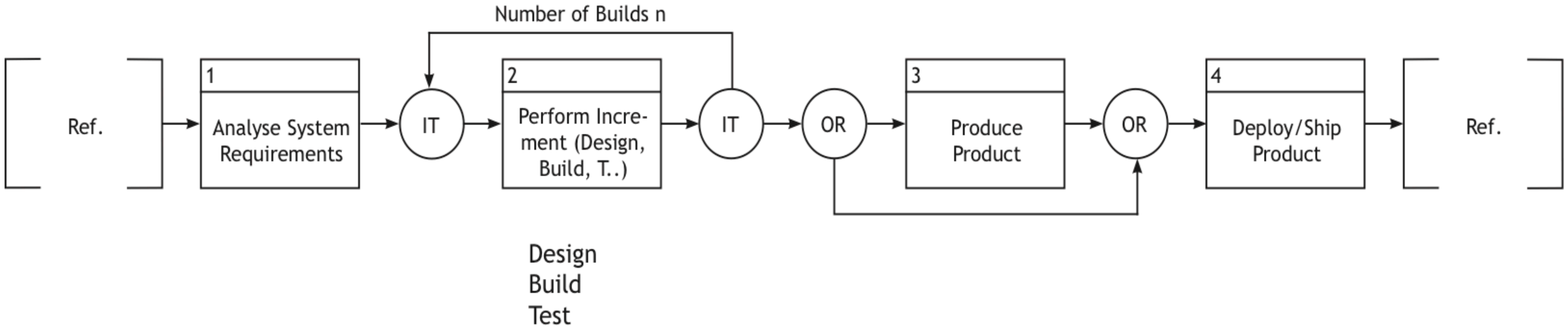
Do:

•Use sequential development (waterfall, grand design, "big bang", etc) for development, where requirements (etc.) are able to be well defined and stable, and solutions are relatively simple or well understood, i.e. the risk due to technology & complexity are low

Why?

- Sequential development is the lowest cost, shortest timeframe approach to development in these circumstances

INCREMENTAL DEVELOPMENT (1)



Note: may be a loop rather than an iteration

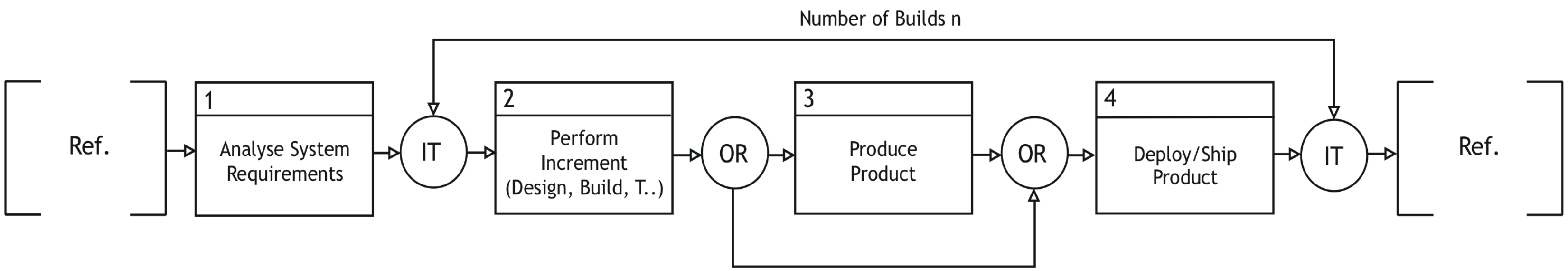
Do:

- Use incremental development where requirements (etc.) can be well defined and stable, but solutions have risk due to technology and/or due to complexity

Why?

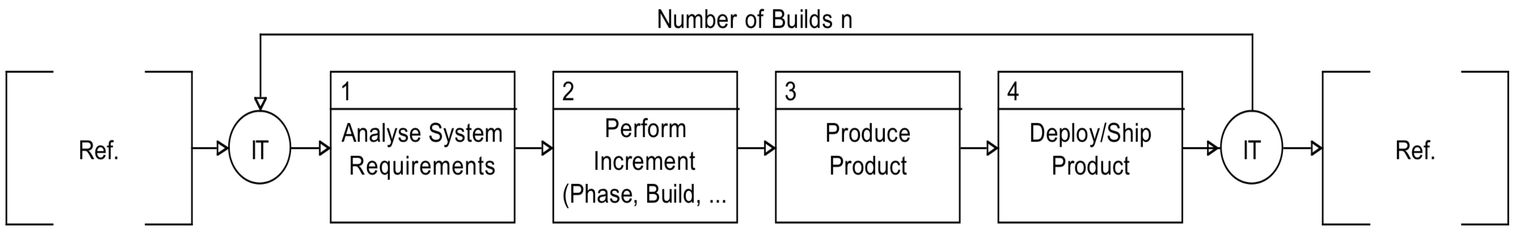
- Incremental development reduces the amount at stake in any build, and allows developers to apply what they have learned to subsequent builds

INCREMENTAL DEVELOPMENT (2)



Note: may be a loop rather than an iteration

EVOLUTIONARY DEVELOPMENT



Do:

• Use evolutionary development where requirements (etc.) are as well-defined as is possible in the circumstances, but remain inadequately defined from the point of view of the company, or are subject to change that the enterprise needs to accommodate

Why?

- Evolutionary development is most able to satisfy end-use needs at the time of supply

Note: evolutionary development should not normally be used as an alternative to capturing what is already known or knowable about requirements – always do that!

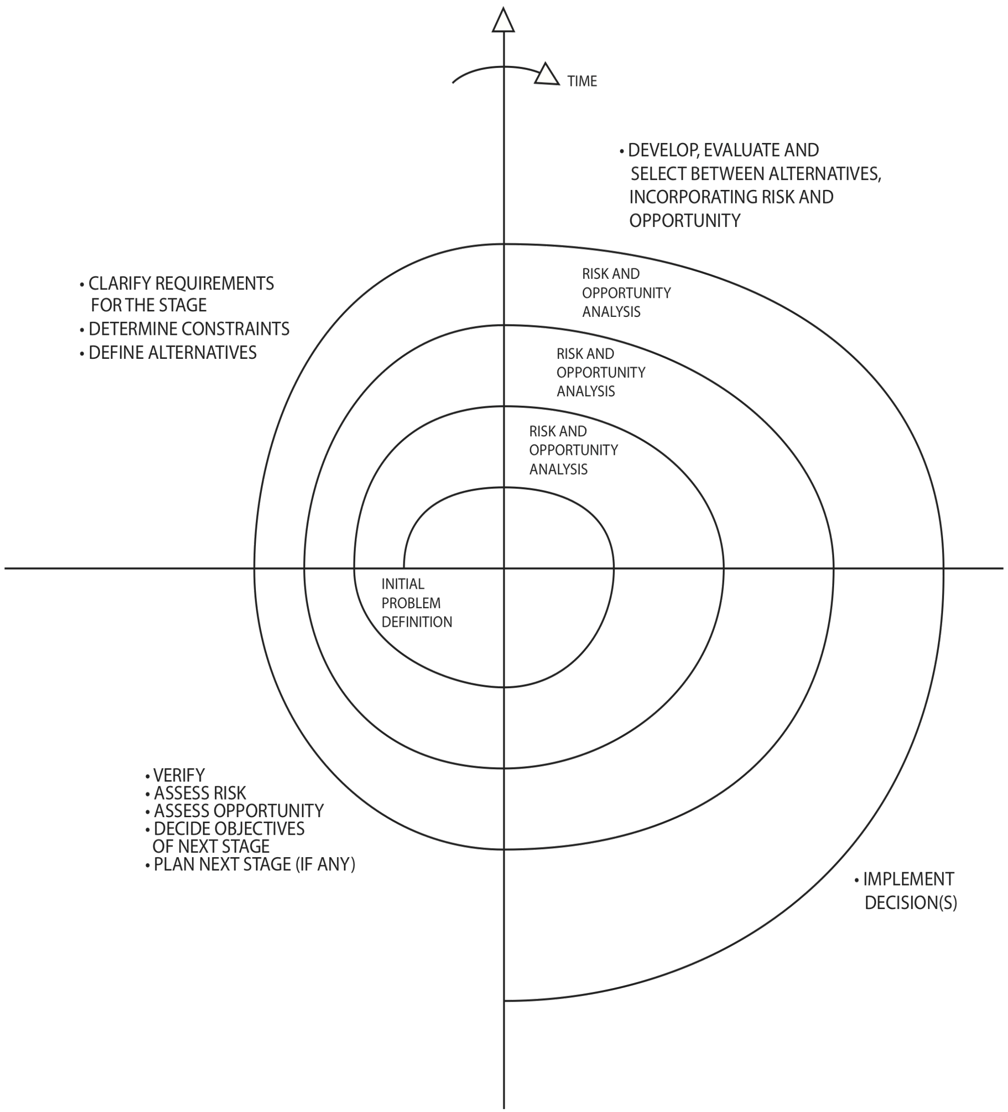
Do:

•Use a stage-based, stage gate, risk and opportunity-driven style of development as an overall strategy for development (sometimes referred to a Spiral development)

Why?

- A stage-based, stage gate, risk and opportunity-driven style of development
  maximises expected value delivered by a project to an enterprise

ILLUSTRATION OF THE SPIRAL MODEL



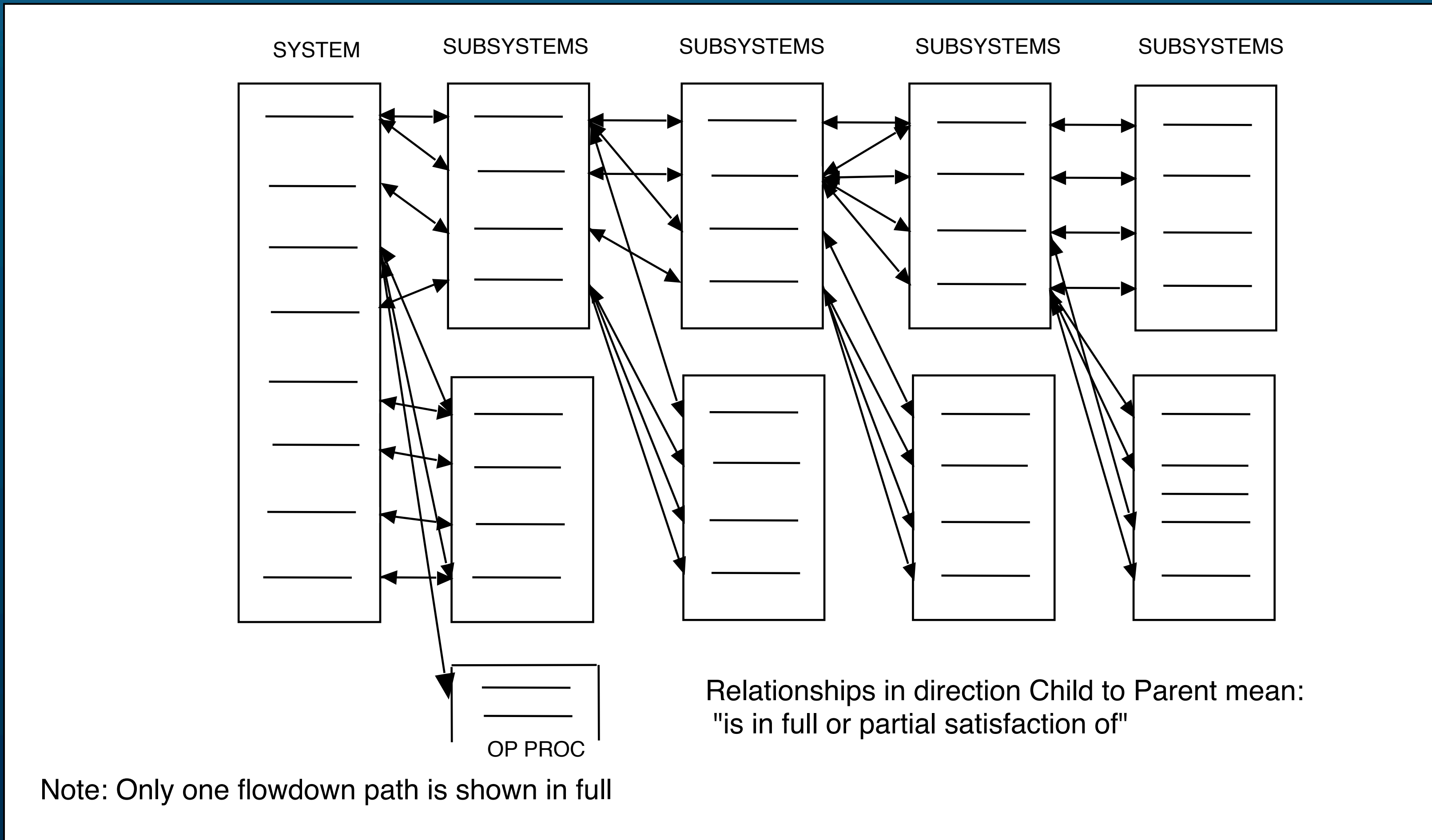
Do:

•Apply the systems engineering process elements selectively within the context of sequential, incremental, evolutionary and/or the risk and opportunity-driven styles of development. Design the development process to match the nature of the problem, using the SE process elements as building blocks

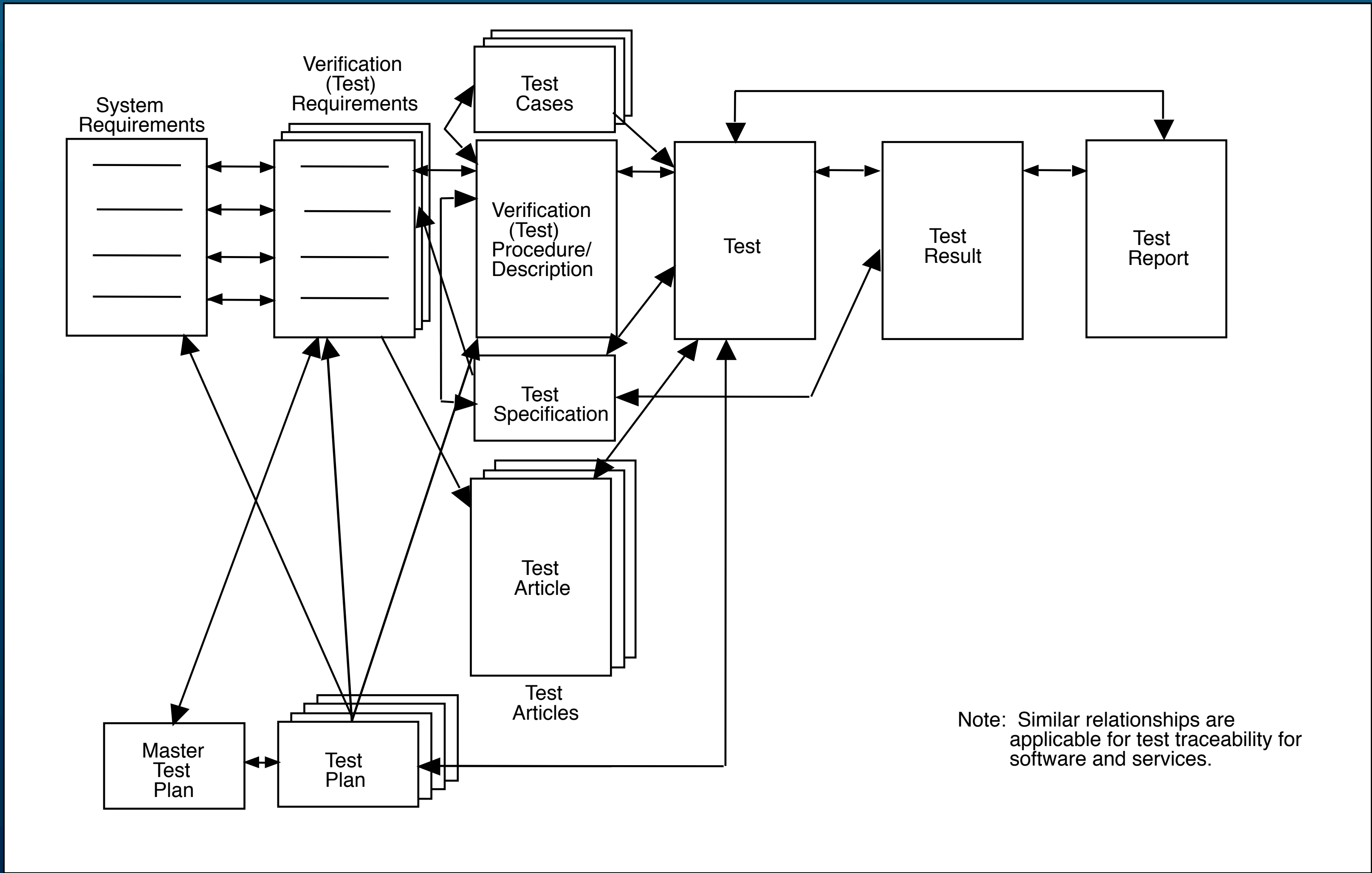
Why?

- Doing everything all of the time is a recipe for overkill. Doing nothing all the time is a recipe for disaster

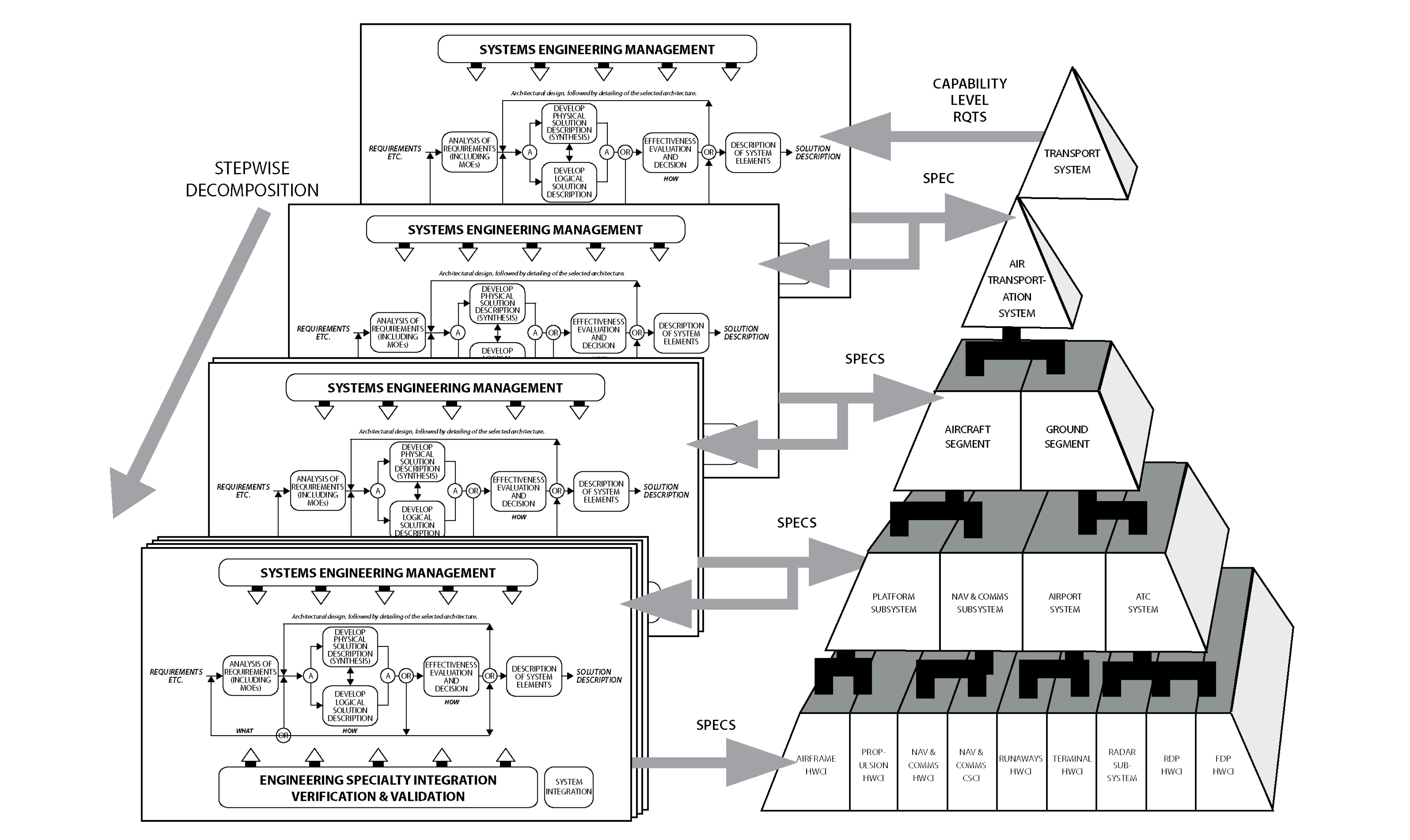
IMPLEMENT REQUIREMENTS TRACEABILITY IN DESIGN



IMPLEMENT VERIFICATION TRACEABILITY



APPLY SYSTEMS ENGINEERING RECURSIVELY ON PROGRESSIVELY SMALLER DEVELOPMENTAL ELEMENTS



Do:

•Maintain a distinction between the statement of the problem to be solved, and the description of the solution to that problem, for the system-of-interest, and for each element of the evolving solution

Why?

- If we don't, and the problem changes, we do not know what we can change and what we cannot change. If we don't, and we need to change the solution, we do not know what we can change and what we cannot change. If we don't, we have lost any reference for verification of solution

Do:

•Baseline (establish a reference definition of) the statement of the problem to be solved, and the description of the solution to that problem. Control changes to requirements (etc.) and to design, maintaining traceability to the applicable baseline

Why?

- Provides a reference for:
  - Acquiring to
  - Supplying to
  - Designing to
  - Verifying to
  - Marketing to
  - Building to

Do:

•Identify and develop solution alternatives that are both feasible (i.e. can meet requirements) and are potentially the most effective

Note: MOEs could include development cost, unit cost of production, time-to-market and other measures unrelated to capability of the product when used

Why?

- Going directly to a point solution may deny the enterprise a much better solution. It the expected benefit exceeds expected cost of extra work, we should do the extra work

Do:

- Develop solutions for relevant enabling systems concurrently, and in balance with, the solution to the system of interest – practice concurrent engineering

Note: An enabling system is a system which makes possible the creation, or ongoing availability for use, of the system of interest during some part of its life cycle, e.g. a production system, a maintenance system

Why?

- Developing the system of interest and enabling systems in sequence results in high costs and long timelines. Decisions are "stovepipe", resulting in rework, or irreversible decisions that compromise capability. This can seriously damage an enterprise

CONCURRENT/SIMULTANEOUS ENGINEERING

- Concurrent
- Collaborative
- Balanced

development of

System of Interest

+

Enabling System

1 or more

e.g. Engineering

System

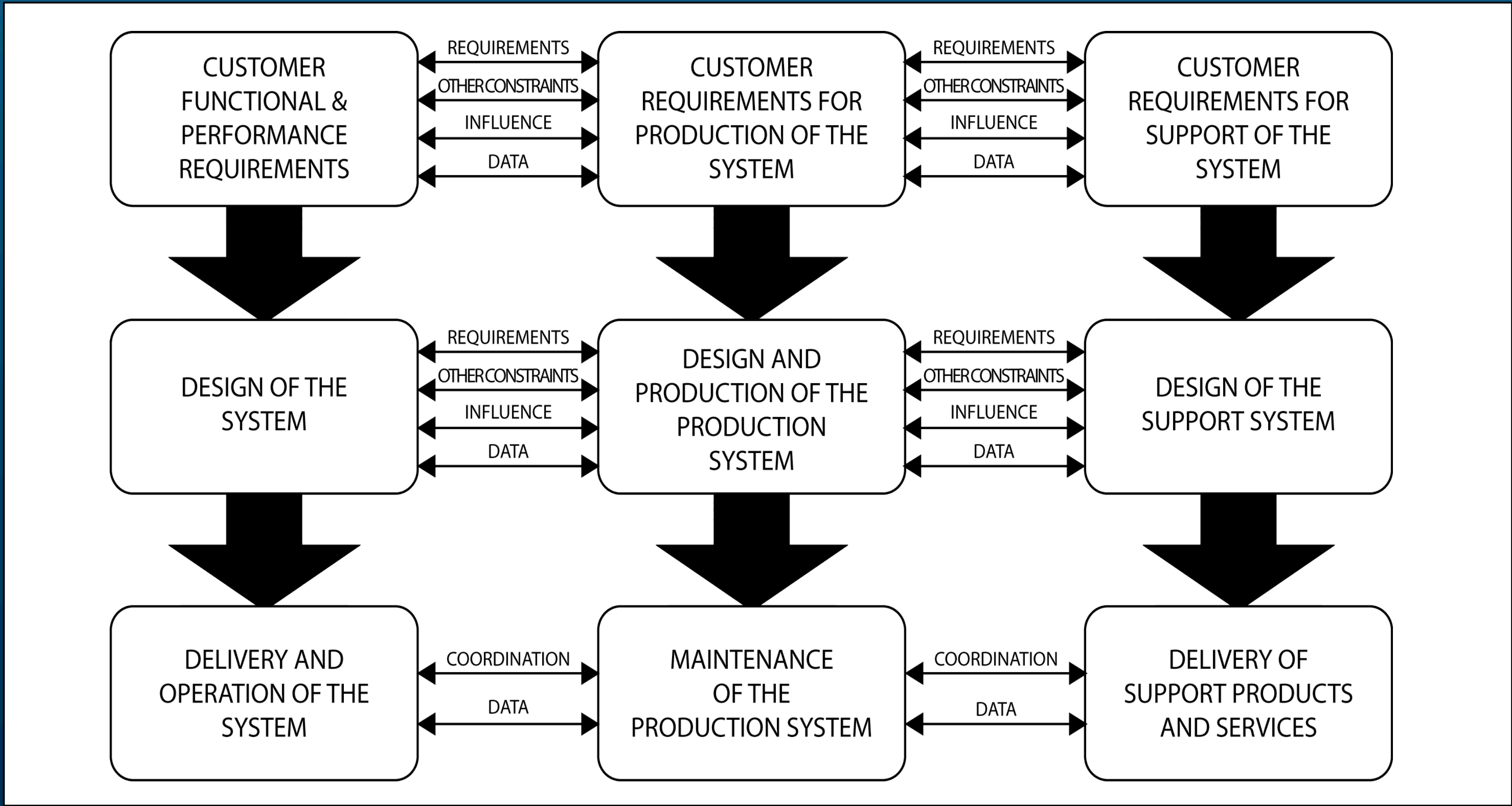
Production System

Maintenance

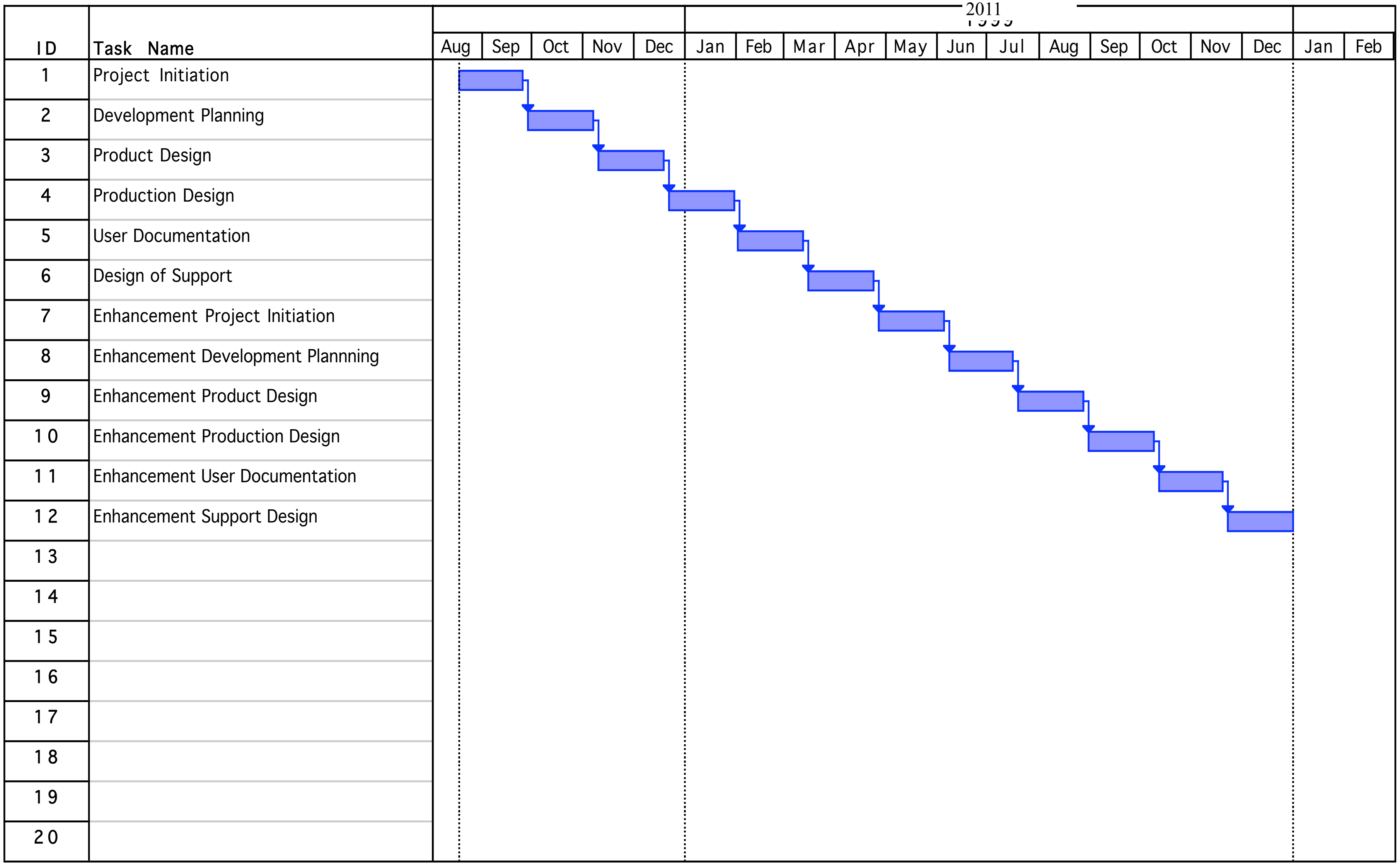
System

Disposal System

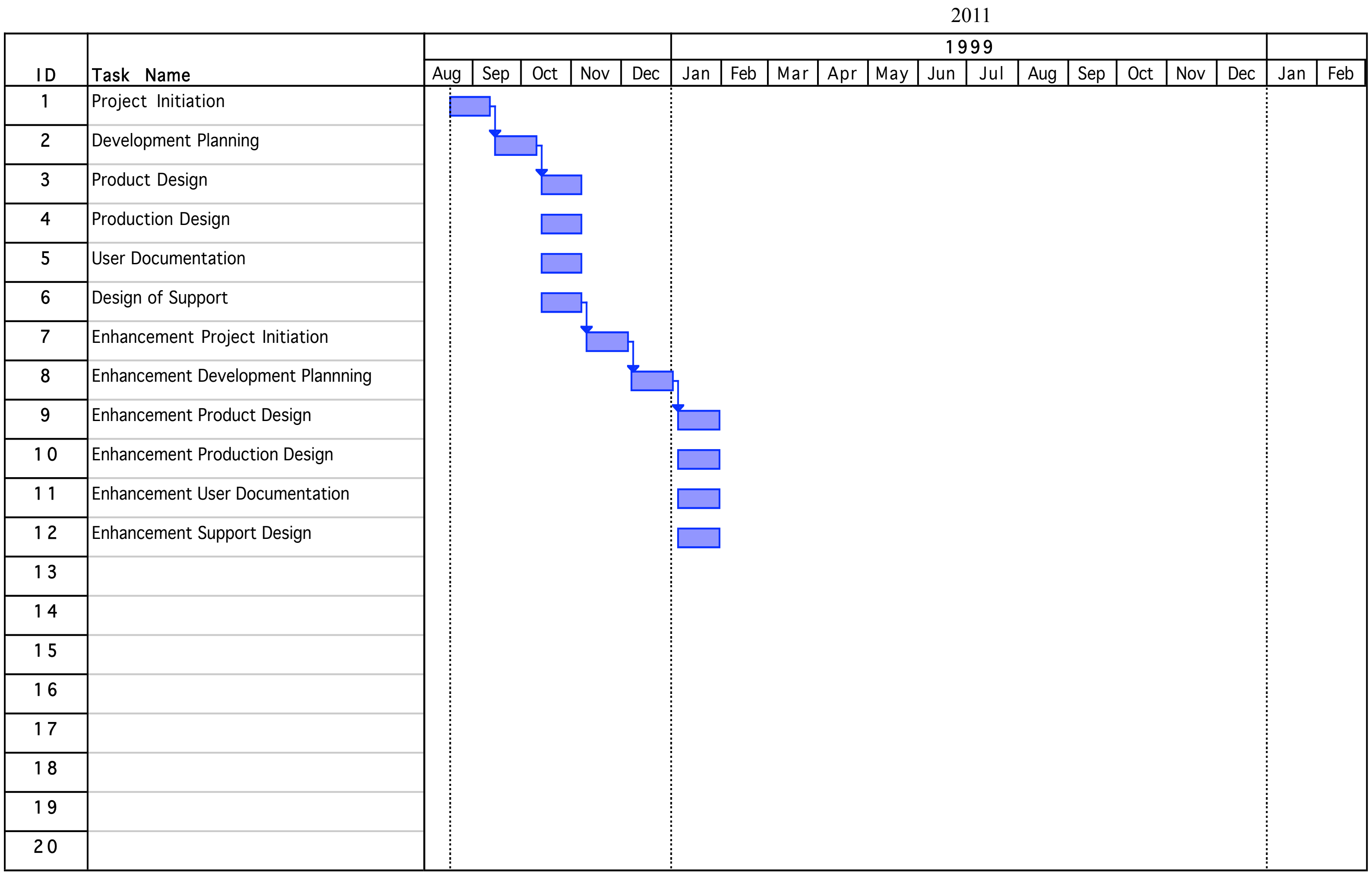
CONCURRENT ENGINEERING CONCEPTS



SEQUENTIAL DEVELOPMENT TIMELINE



CONCURRENT DEVELOPMENT TIMELINE



Concurrency shortens the timeline

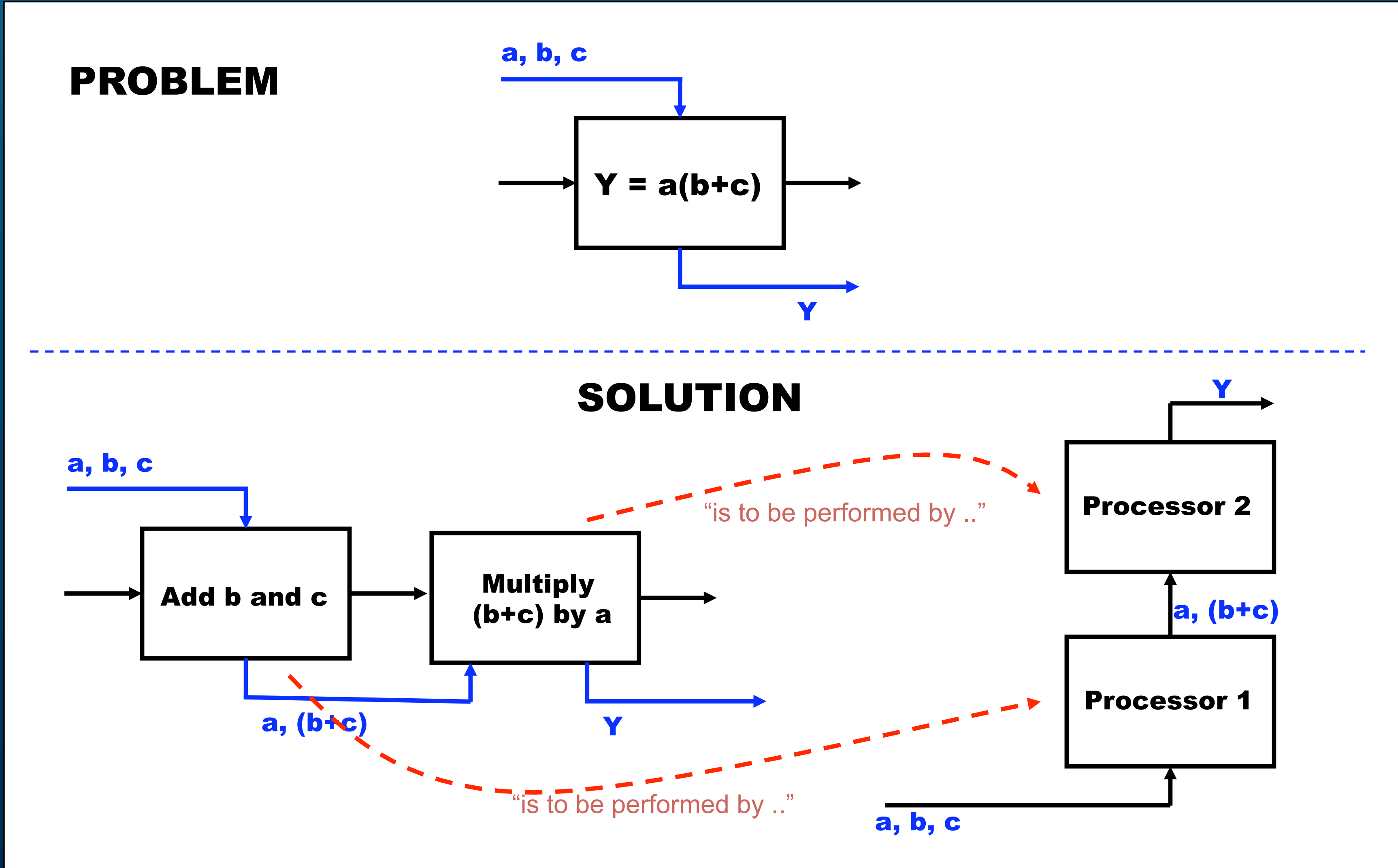
Do:

•Except for simple problems, develop logical solution descriptions (description of how the system is to meet its requirements) as a means of developing physical solution descriptions (description of how to build the system)

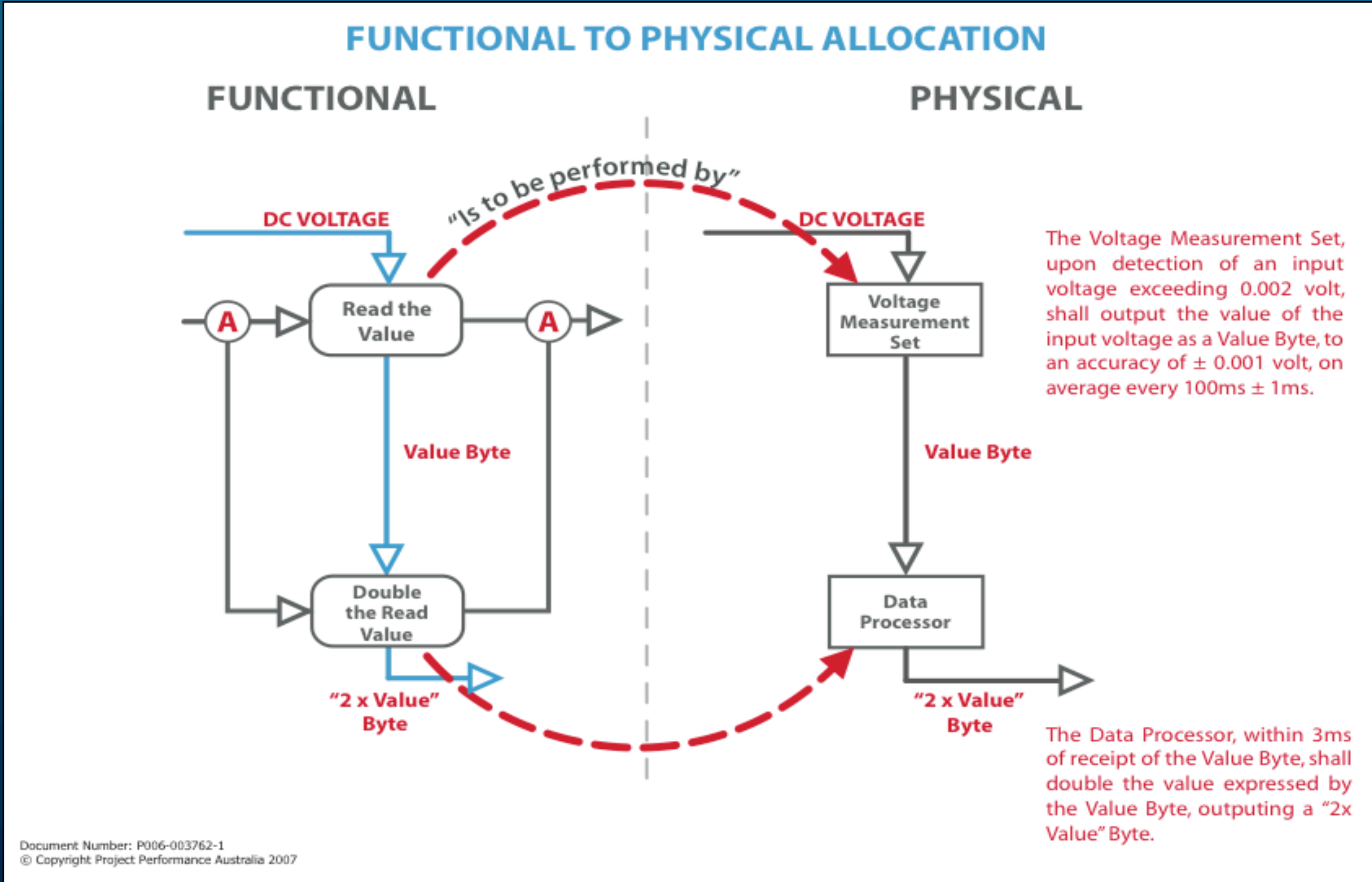
Why?

- For simple problems, the cost of formalizing the logic will exceed the benefit. For other problems, the avoided cost of rework due to design errors will more than justify the cost in time and money of the extra work

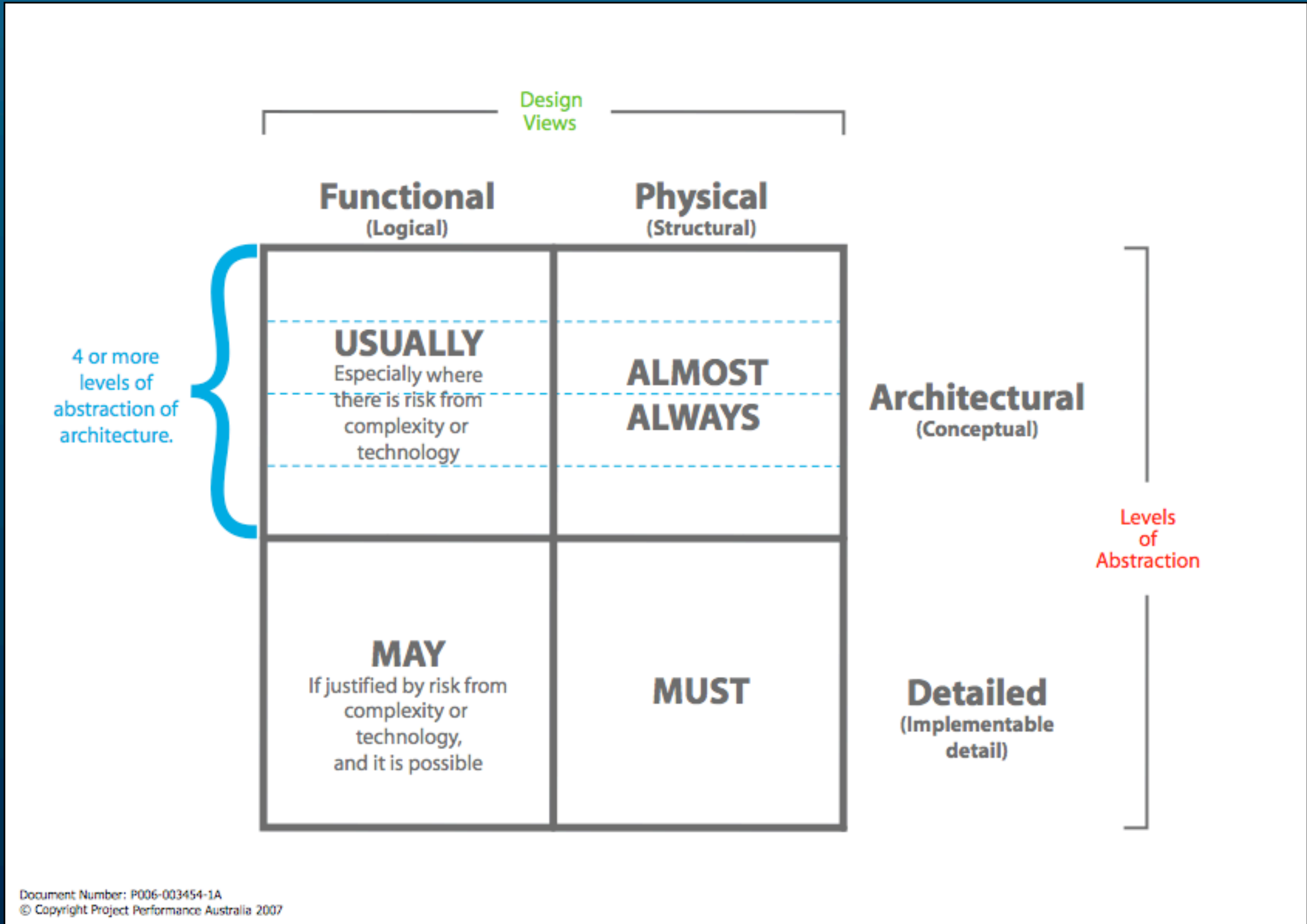
PHYSICAL AND LOGICAL DESIGN - EXAMPLE



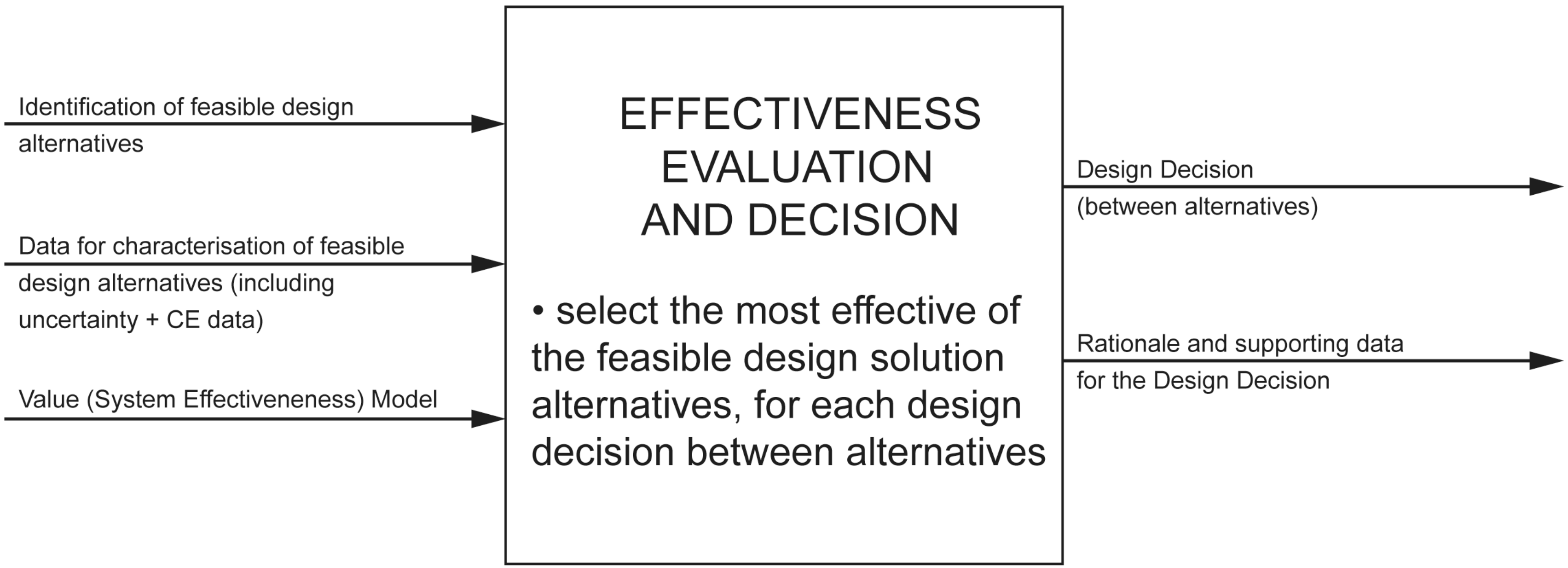
PHYSICAL AND LOGICAL DESIGN - EXAMPLE



DESIGN VIEWS - PHYSICAL AND LOGICAL



EFFECTIVENESS EVALUATION AND DECISION



Legend
- CE concurrent engineering

Do:

•Select between (feasible) design alternatives based on the evaluation of risk-adjusted expected benefit to applicable stakeholders, i.e., on expected overall effectiveness

Note: “expected effectiveness” refers to effectiveness which incorporates uncertainty, reflecting risk and opportunity

Why?

- This strategy will produce the best average outcome from our engineering

Do:

•Be prepared to iterate in design, to drive up the benefit to the applicable (primary) stakeholders of the outcomes of design

Why?

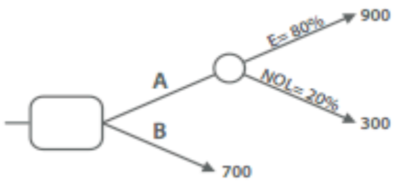
- To do otherwise is to assume that, as designers, we always come up with the best, for our enterprise, implementation of a concept the first time. This is rarely the case

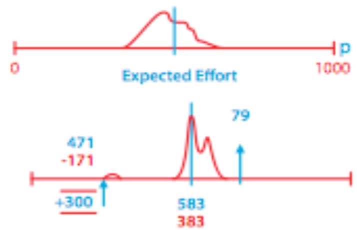
TRADE STUDIES AND DESIGN ITERATION

ARCH. A

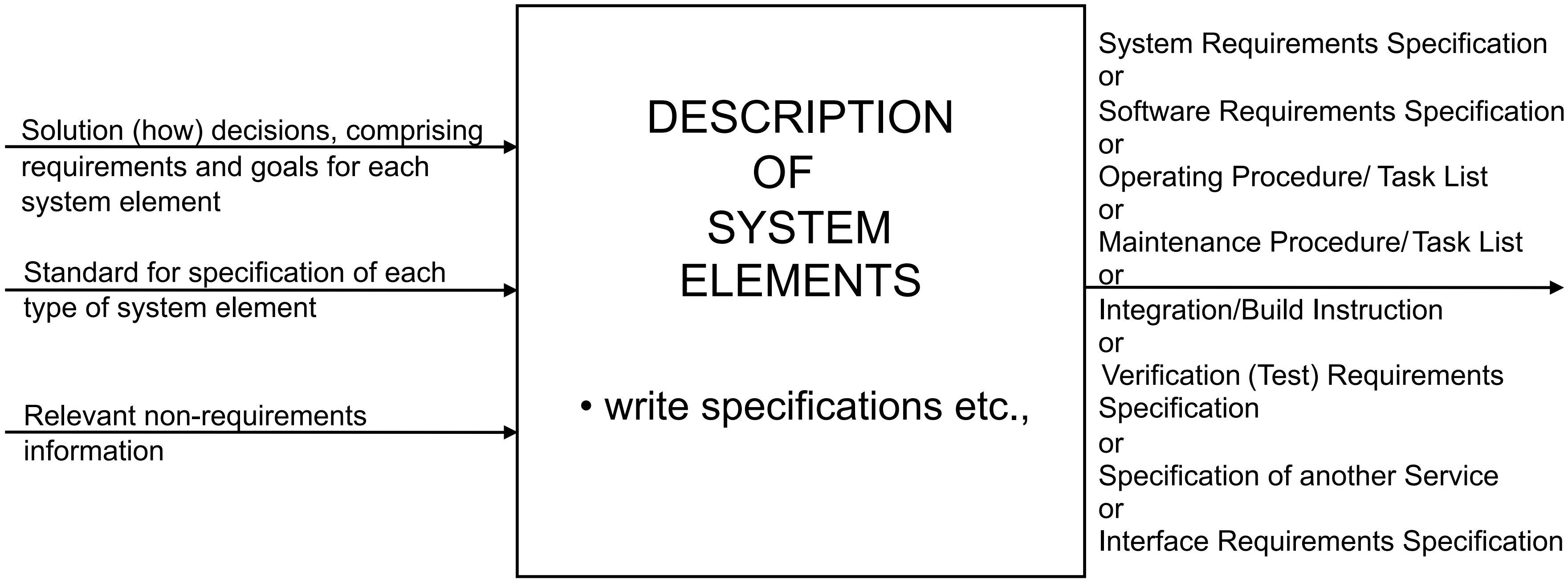
Moes	Worst	Best	Pri	pts	%	UF	Value of MOE	RVC	AVC (RVC x wt)	
Cost, $K's p/u	200k	50k	1	100	25		~~55k~~ 57	10	250	
Reliability %	95	100	1	100	25		~~95.5%~~ 97	~~100~~ 5	~~25~~ 125	+100
Intercoperability	0	17	7	14	4		0	0	0	0
Size(A/B/C)	C	A	8	3	1		~~C~~ B	~~0~~ 5	~~0~~ 5	+5
Scheldule (Month)	12	6	3	40	10		~~7~~ 8	~~10~~ 9	~~100~~ 90	-10
Visible optical range	1000	5000	5	30	7		~~1200~~ 2500	~~2~~ 5	~~14~~ 35	+21
Duration of transmission, hr	48	96	6	27	6		50	0.5	3	
Readiness %	90	100	4	39	10		~~91~~ 95	~~1~~ 5	~~10~~ 50	+40
OS & d cost $K pu/10 years	300	10k	2	50	12		~~200k~~ 106	~~1.5~~ 8	~~18~~ 96	+74
				392	100			Σ	420	567

VM

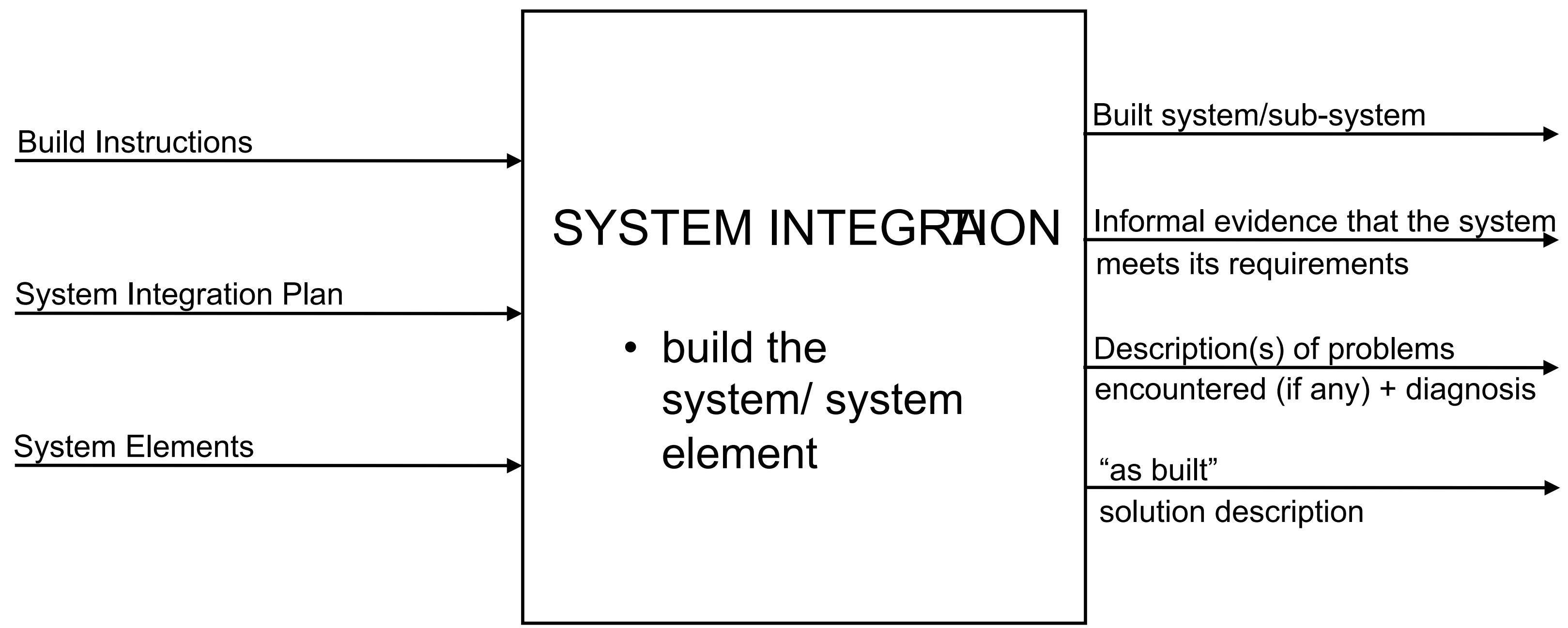




DESCRIPTION OF SYSTEM ELEMENTS



SYSTEM INTEGRATION



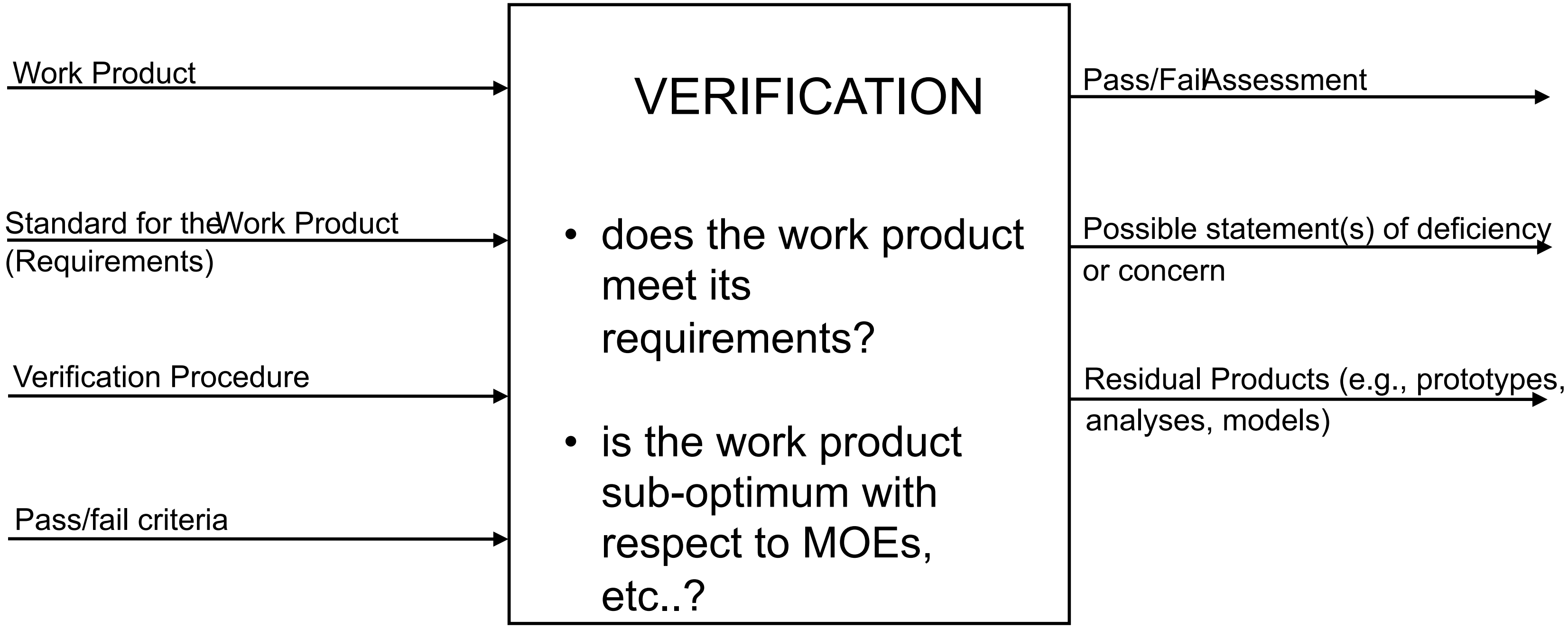
Do:

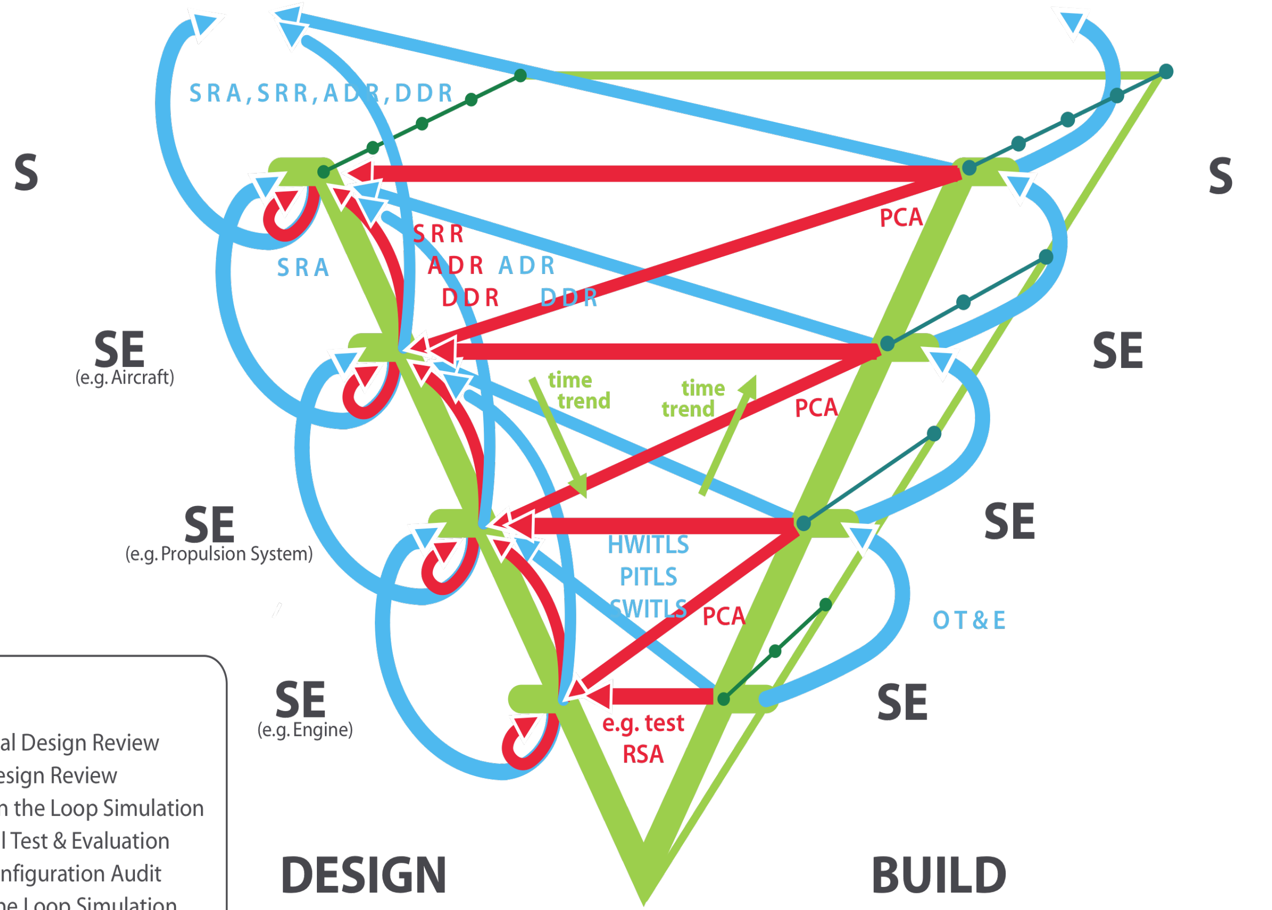
•Subject to level of risk, independently verify work products (is the job being done right, i.e., does the work product meet the requirements for the work product?)

Why?

- Verification is a risk-reduction activity. If the amount of risk reduction exceeds the cost of the verification activity, it is a good thing to do

VERIFICATION





Legend :

●	A Build
ADR	Architectural Design Review
DDR	Detailed Design Review
HWITLS	Hardware in the Loop Simulation
OT&E	Operational Test & Evaluation
PCA	Physical Configuration Audit
PITLS	People in the Loop Simulation
RSA (FCA)	Requirements Satisfaction Audit
S	Top Level System
SE	System Element
SRA	System Requirements Analysis
SRR	System Requirements Review
SWITLS	Software in the Loop Simulation

Document Number: P006-003758-3

Verification :
Is the work product correct-meets requirements?

Validation:
Does the work product satisfy the need for the work product?

WEDGE DIAGRAM™

Don’t:

•Rely on technical progress meetings with the customer for design verification, even if these meetings go under the name of “design reviews”

Why?

- The cost of correction of design errors undiscovered in design verification exceeds the cost of (discovery + early correction) by a factor of about 5:1

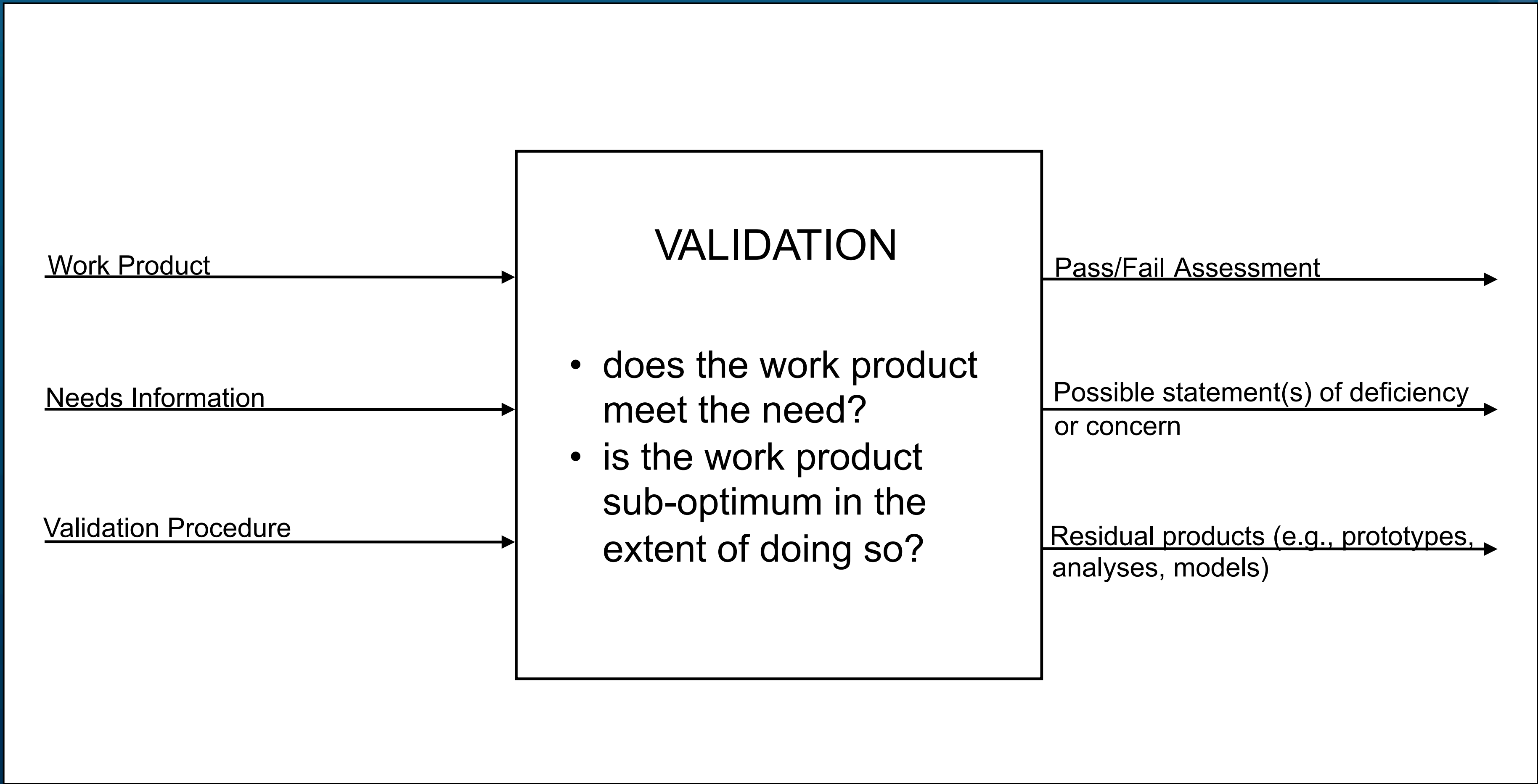
Do:

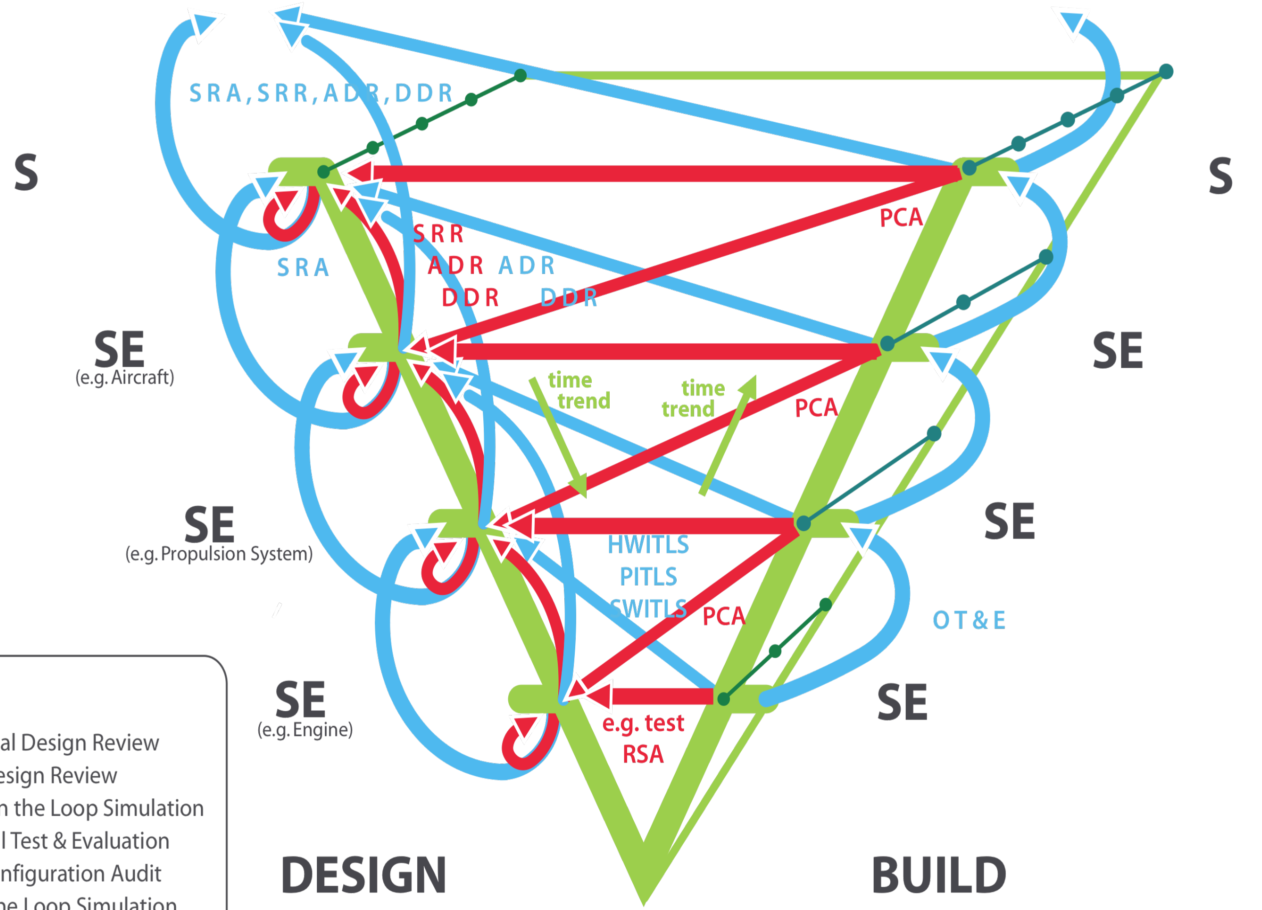
•Subject to level of risk, independently validate work products (is the right job being done, i.e. does the work product meet the need for the work product?)

Why?

- Validation is a risk-reduction activity. If the amount of risk reduction exceeds the cost of the validation activity, it is a good thing to do

VALIDATION


Work Product
Needs Information
Validation Procedure
VALIDATION
• does the work product meet the need?
• is the work product sub-optimum in the extent of doing so?
Pass/Fail Assessment
Possible statement(s) of deficiency or concern
Residual products (e.g., prototypes, analyses, models)



Legend :

●	A Build
ADR	Architectural Design Review
DDR	Detailed Design Review
HWITLS	Hardware in the Loop Simulation
OT&E	Operational Test & Evaluation
PCA	Physical Configuration Audit
PITLS	People in the Loop Simulation
RSA (FCA)	Requirements Satisfaction Audit
S	Top Level System
SE	System Element
SRA	System Requirements Analysis
SRR	System Requirements Review
SWITLS	Software in the Loop Simulation

Verification :
Is the work product correct-meets requirements?

Validation:
Does the work product satisfy the need for the work product?

WEDGE DIAGRAM™

Document Number: P006-003758-3

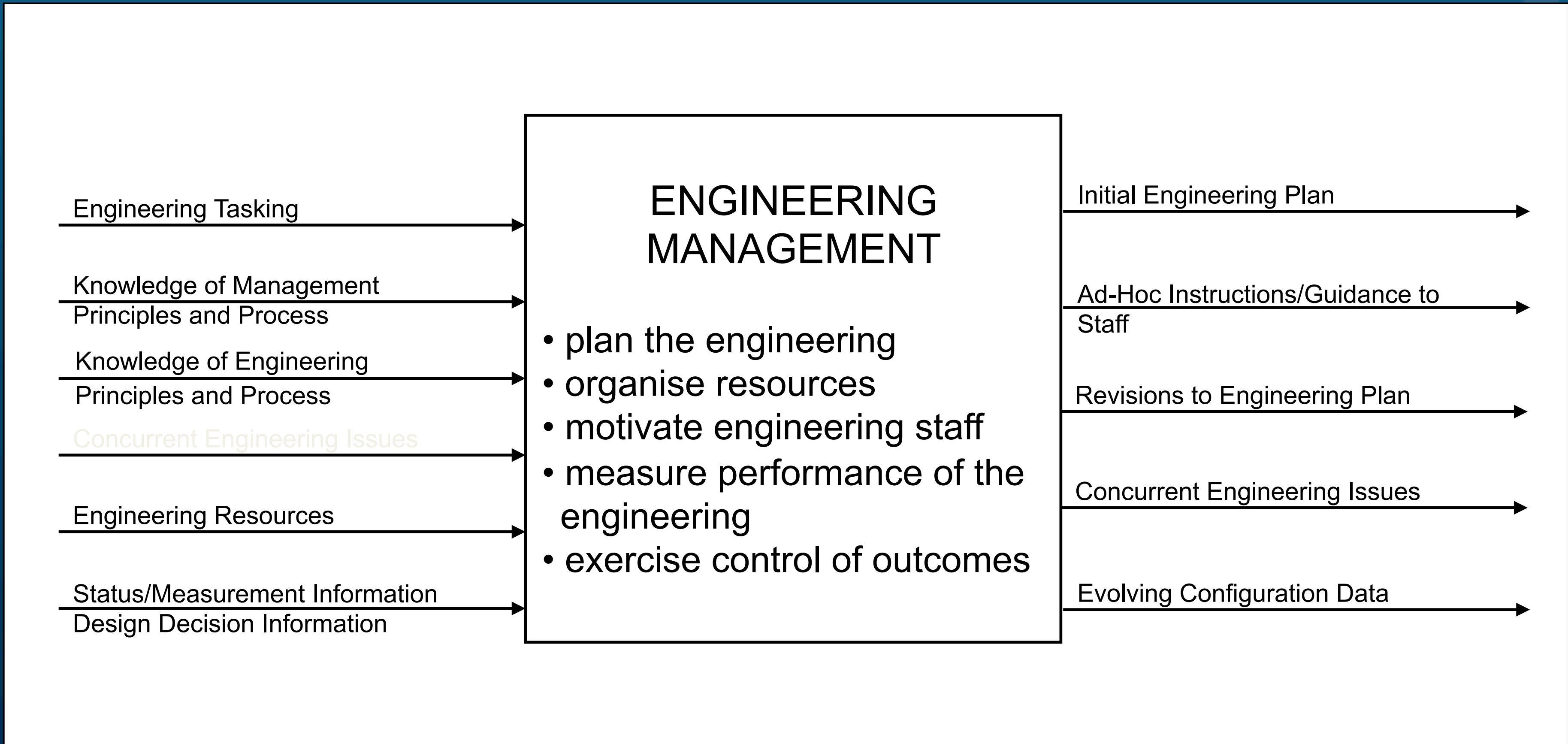
Do:

- Manage the engineering – plan, organize, motivate, assess, control

Why?

- Studies show a 7% increase in return on sales between companies that routinely plan and control their projects versus those that don't

ENGINEERING MANAGEMENT



Do:

•Recognize that the engineering system is a system like any other system. Engineer it as such

Why?

- Engineering the engineering system aims to produce an optimum implementation of the engineering system, with all work done adding maximum value, compared with the alternatives, for the enterprise

Do:

•Use a product-oriented structure of products and related services (PBS/ WBS) as a framework for definition, cost estimating, scheduling, risk analysis, measurement, assignment of responsibility, team design and reporting

Why?

- PBS/WBS is an enormously powerful tool in managing the engineering and the project. But to be a powerful tool, it must be developed within a set of principles and rules

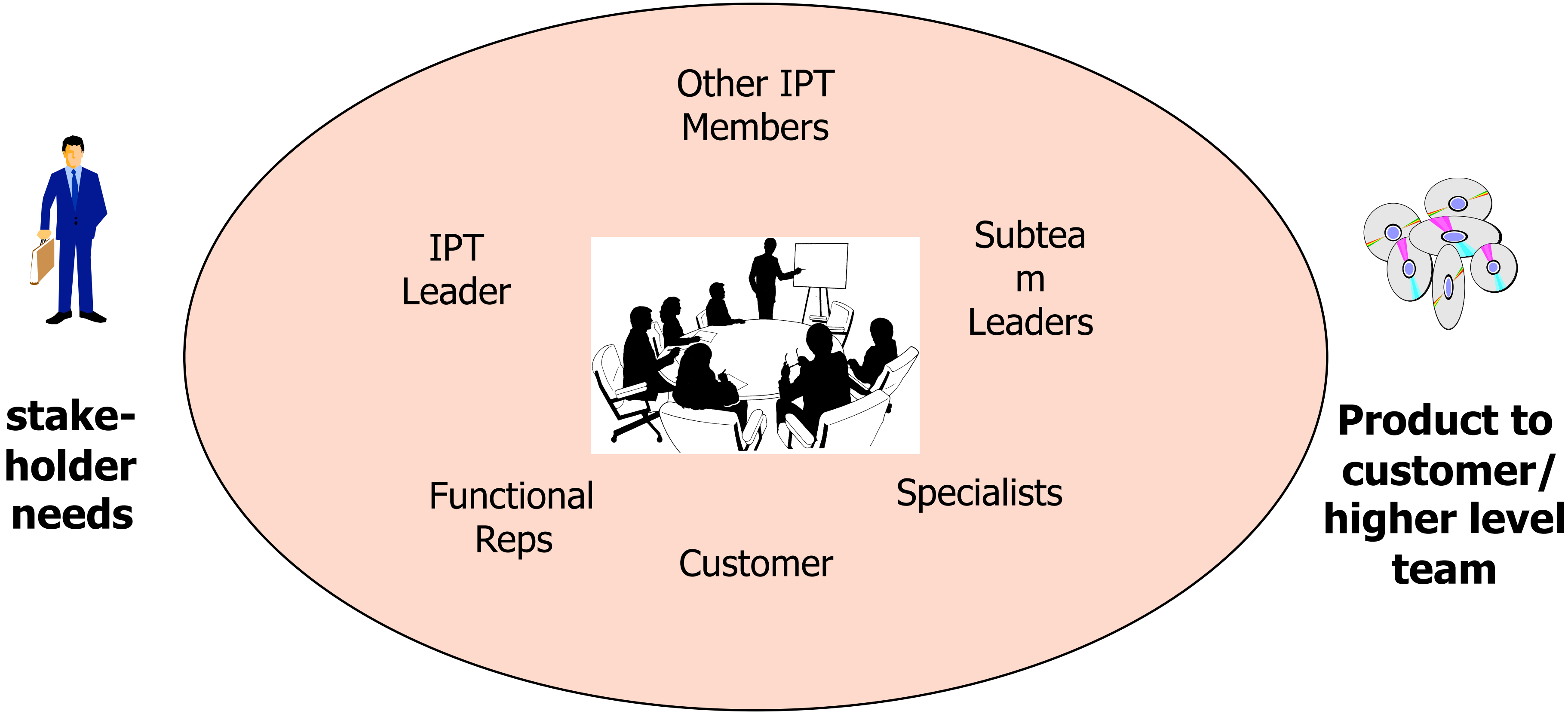
Do:

- Use empowered, product-oriented, multidisciplinary team structures for larger engineering efforts

Why?

- Integrated Product Teams (IPTs) have a well-established record of higher performance than alternative organizational units

INSIDE AN INTEGRATED PRODUCT TEAM



- A *multi-disciplinary, cross-functional, stake holder-focussed* team solely responsible for taking a product from need to delivery
- Knowledge, skills and attitudes of the team members are complementary

Do:

- Choose to do things only in the rational expectation of producing a better result by doing so (on the balance of probabilities). Choose to NOT do things for EXACTLY the same reason

Why?

- Because to do otherwise is to set out to achieve a worse result for the enterprise, and that's crazy!

Don't:

- Defy history without having a very good reason for doing so - adopting courses of action that have historically failed

Why?

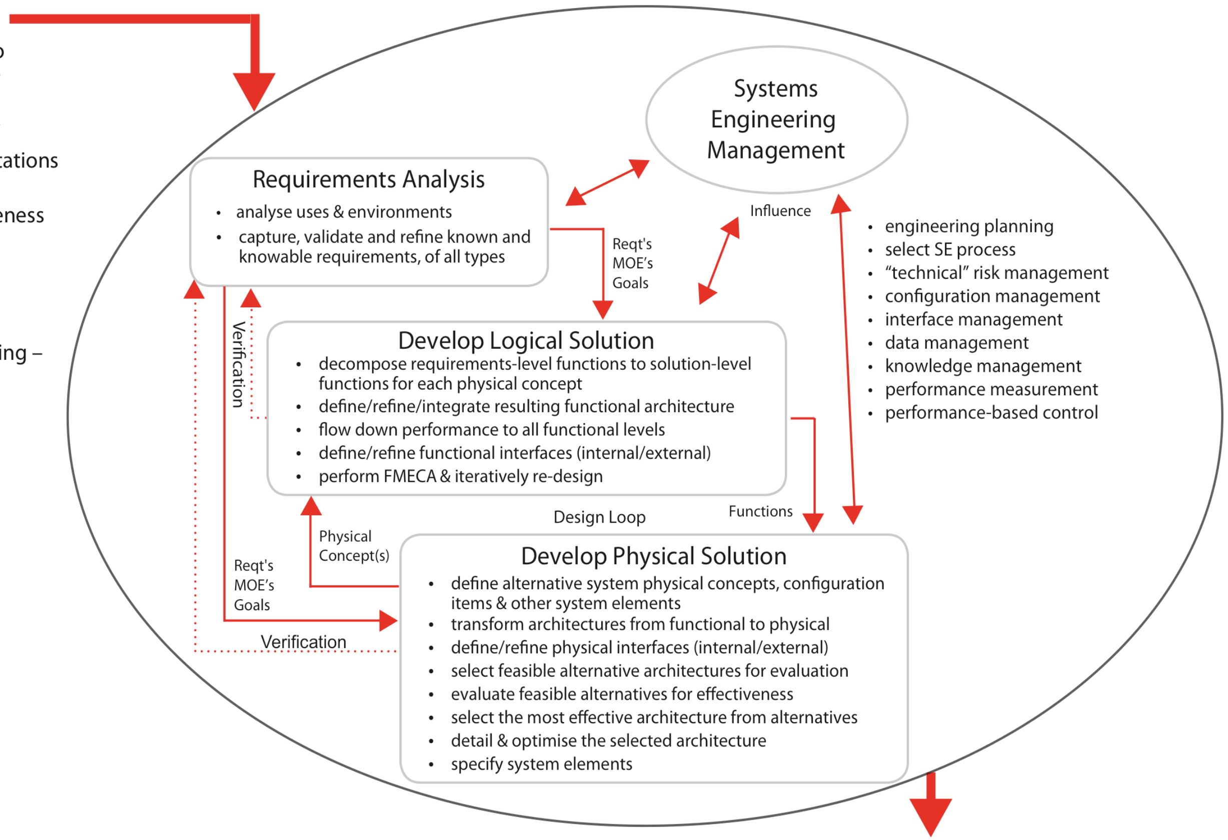
- The past is a good pointer to the future

PROJECT PERFORMANCE INTERNATIONAL

www.ppi-int.com

PROCESS INPUT

- problem domain info
- user/customer/other stakeholder needs/ desires/wants/goals/ requirements/expectations
- uses/missions
- measures of effectiveness
- value information
- environments
- other constraints
- technology base
- concurrent engineering – related inputs



PROCESS OUTPUT

- **identification & specification of each system element, including build instructions**
- requirements traceability information
- system & system element verification requirements
- design traceability information (decision data base)
  - system functional & physical architecture and detail descriptions
  - design decision support data
  - design decision rationale data
- concurrent engineering-related outputs
- prototypes, where applicable

Note 1: The Systems Engineering Process is applied repeatedly to each design object, starting at, for example, the Capability, Mission or Use System, then to, for example, the Prime Mission or Use Product, Maintenance System, Production System, Operational Infrastructure, etc, then to subsystems of these systems.

Note 2: Also, where applicable, validate data products (not shown diagramatically)

Note 3: The process also controls the integration of the system elements to build the system for the first time (system integration).

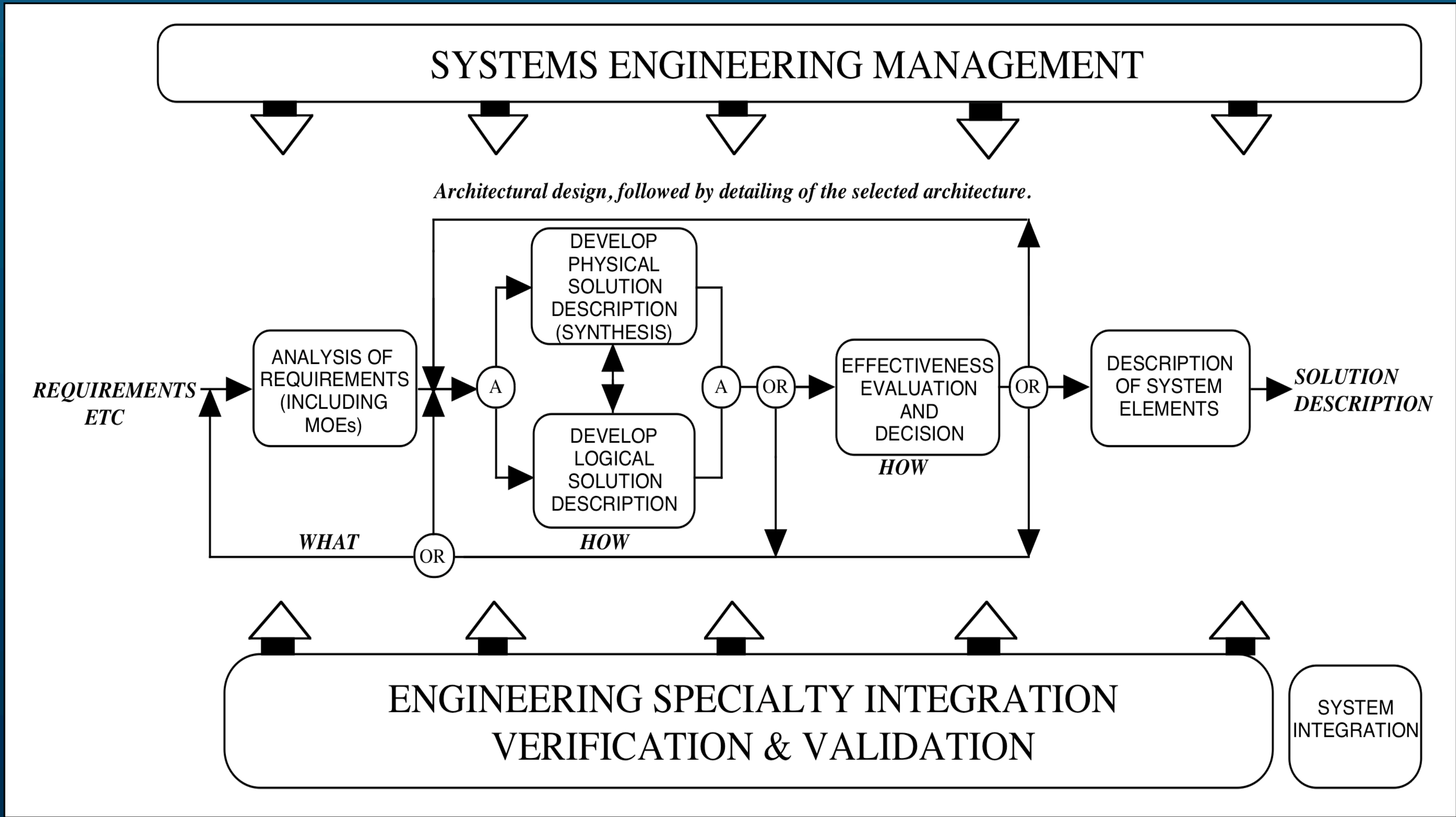
Note 4: The process also includes the conduct of verification of the produced system against the requirements for that system, therebyverifying both the system, and the design of the system.

Note 5: The process also includes the conduct of validation of the produced system against the need.

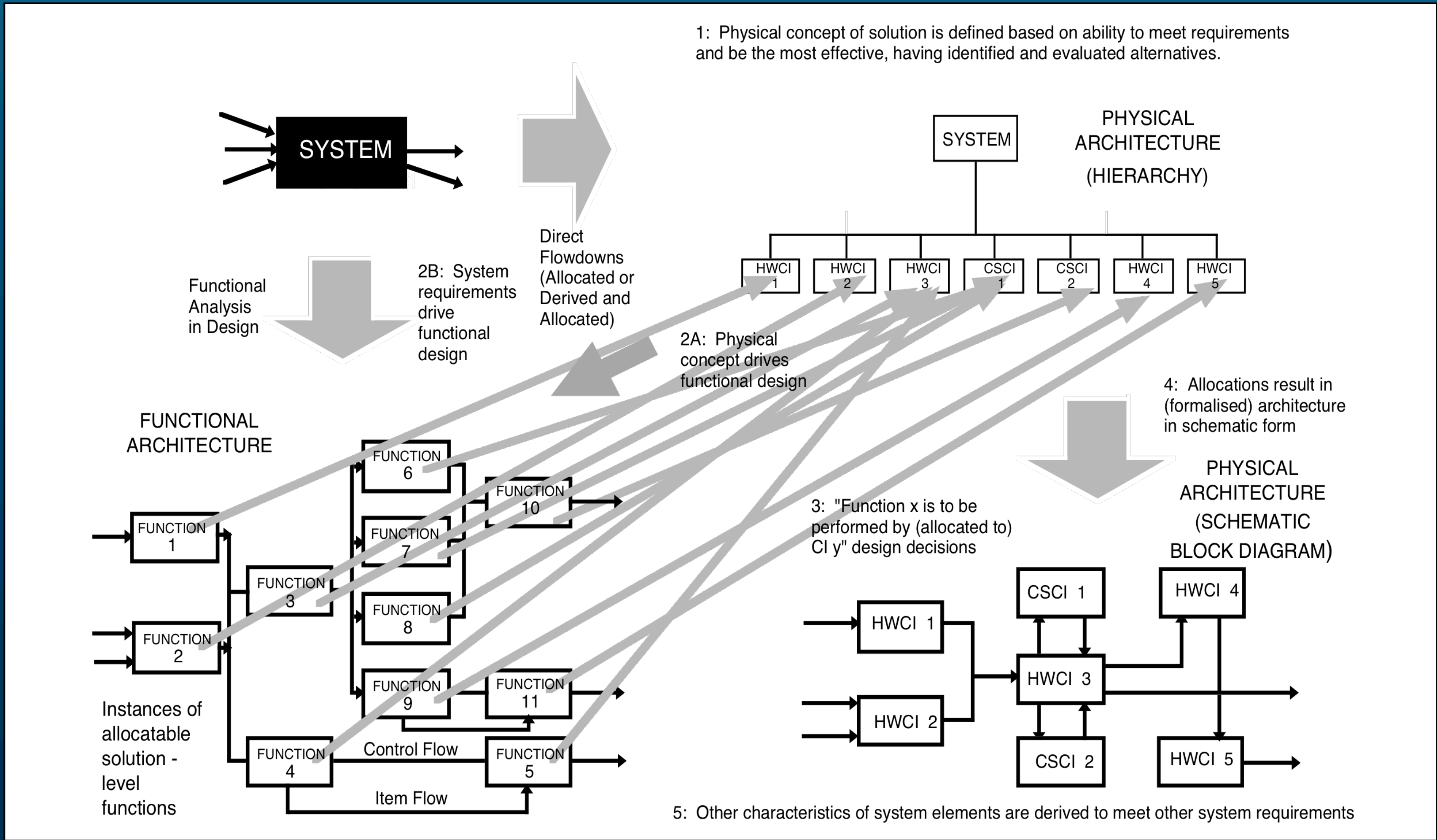
A Systems Engineering Process View

P006-002827-32

SYSTEMS ENGINEERING – BASIC PROCESS ELEMENTS



ALLOCATION OF FUNCTIONS TO ARCHITECTURAL ELEMENTS



THE ROLE OF COGNITIVE SYSTEMS ENGINEERING

Cognitive Systems Engineering (CSE) is an approach to the design of technology, training, and processes intended to manage cognitive complexity in sociotechnical systems

Militelo, Dominguez, Lintern and Klein, “The Role of Cognitive Systems Engineering in the Systems Engineering Design Process”, Systems Engineering, Vol 13, No. 3, 2010

WORKSHOP 1 - PRINCIPLES OF THE ENGINEERING OF SYSTEMS

The Objective

To consider a set of principles which may be applied in the engineering of systems (of all types)

The Task

1. Carefully read the handout, titled "Systems Engineering Principles". Then consider as a group, for each principle, the following questions:

    a. Is it a valid and beneficial principle in the performance of our engineering?

    b. Do I have any questions, issues or qualifications?

In the wrap-up, a person who has carriage of a principle for the group should read out the principle aloud, present the group's conclusion, and then invite comment. Quickly hand over to the next group/person/principle when useful discussion has concluded, or if no useful comment is forthcoming

Your facilitator will be available to answer questions and correct any misunderstandings of intent of the statements of the principles

CAPABILITY MATURITY MODELS

Reference models against which engineering-related capability may be assessed:

- EIA-731 - good
- CMMI - much very good content, but problems in requirements management, requirements development & technical solution development
- ISO/IEC TR 15504
- Other CMMs

LOGISTIC SUPPORT ANALYSIS

- Would better be titled “Logistic Support Analysis and Design”
- It is to the support solution what end use product design is to the end use product solution - it is a systems engineering approach applied to the design of the support system, and its interface with the products which are to be supported

KEY SYSTEMS ENGINEERING ARTIFACTS

- Systems engineering plans
- Operational concept descriptions
- System requirements specifications
- Interface requirements specifications
- Verification requirements specifications
- Architectural design descriptions
- Detailed design descriptions
- Test/verification procedures
- Records of test/verification results
- Validation procedures
- Records of validation results

SYSTEMS ENGINEERING PLAN

Description: The Systems Engineering Plan (SEP) defines the plans and procedures of an enterprise for the management and conduct by that enterprise of a fully integrated technical program in conduct of the engineering element(s) of a project or a part thereof. The term "SEP" is generic, and may be replaced with any meaningful name. The term "enterprise" may be interpreted to mean any entity responsible for performance of the work which is the subject of the SEP. The SEP, including or supplemented by subordinate plans, is used to provide the primary work planning and process direction and guidance to the technical team responsible for conduct of the work. The SEP may also be used to provide visibility, to a customer, of a supplier's engineering planning and intended processes. The content of the SEP is intended to be responsive to contract requirements, if any, but is not, itself, intended to be invoked contractually

Acronyms: SEP, SEMP, EMP

Standards: PPA-ME04-000905 current release, DI-MGMT-81024

OPERATIONAL CONCEPT DESCRIPTION

Description: The Operational Concept Description (OCD) describes, for a system, subsystem, HWCI, CSCI, component or other item, herein referred to generically as "the system", who the users of the system are, what are the intended uses of the system, how and where the system is intended to be used, and a representative set of scenarios of use. These scenarios, each associated with a particular intended use (mission), are chosen to represent both typical and limit conditions of use. The OCD provides a direct reference for validation of requirements, and fitness for intended use of the solution

Acronyms: OCD, ConUse, ConEmp, SOI

Standards: PPA-ME04-000950 current release

SYSTEM REQUIREMENTS SPECIFICATION

Description: A System Requirements Specification (SyRS) specifies the requirements to be satisfied by a system, subsystem, HWCI, component or other physical item, and optionally the requirements for evidence that each requirement has been so satisfied. Requirements pertaining to the system, subsystem or item's external interfaces may be presented in the SyRS or in one or more Interface Requirements Specifications (IRSs) or Interface Control Documents (ICDs) invoked by reference from the SyRS. The SyRS, possibly supplemented by IRSs or ICDs, is commonly used as the basis for acquisition, supply design and development, verification and acceptance of the system, subsystem or other item

Acronyms: SyRS, SSS, SRS, PDS, FPS, ORD, MRD, A B or C Spec, ..

Standards: PPA-002235 current release

SOFTWARE REQUIREMENTS SPECIFICATION

Description: The Software Requirements Specification (SRS) specifies the requirements to be satisfied by a software item (eg. software system, subsystem, CSCI, component or other item), and, optionally, corresponding verification requirements. Requirements pertaining to the software item's external interfaces may be presented in the SRS or in one or more Interface Requirements Specifications (IRSs) referenced from the SRS. The SRS, possibly supplemented by IRSs, is used as the basis for procurement, design, verification testing and acceptance testing of the software item

Acronyms: SRS, SoRS

Standards: PPA-002237 current release

INTERFACE REQUIREMENTS SPECIFICATION

Description: An Interface Requirements Specification (IRS) specifies the requirements to be satisfied at an interface between two items (hardware-hardware including hardware-human, hardware-software or software-software) and, optionally, corresponding verification requirements. The IRS is used in support of procurement, design, verification testing and acceptance testing of one or both of the items

Acronyms: IRS

Standards: PPA-ME04-002234 current release

INTERFACE DESIGN DESCRIPTION

Description: The Interface Design Description (IDD) describes the design characteristics at an interface between two items (hardware-hardware including hardware-human, hardware-software or software-software). The IDD is used in system development to record, communicate and control external interface design, at the most detailed level of definition of an external interfaces, and consistent with requirements contained within the corresponding Interface Requirements Specification (IRS). The IRS specifies interface requirements; the IDD describes interface characteristics selected to meet those requirements. The IDD can be used to supplement a System/ Subsystem Design Description (SSDD), Software Design Description (SDD), or Database Design Description (DBDD). An IDD may describe one or more interfaces

Acronyms: IDD, ICD

Standards: PPA-004611 current release

VERIFICATION REQUIREMENTS SPECIFICATION

Description: The Verification Requirements Specification (VRS) describes the qualities of the evidence required that a set of requirements defining an item is satisfied. The item may be of any nature whatsoever, ranging from, for example, a physical object, to software, to an interface, to a data item, to a material, or a service. The VRS is used to communicate to verification design personnel the characteristics required of any verification solution, i.e. the VRS is a major input to the development of test procedures and similar. The VRS also provides the criteria against which test, and other verification procedures, may themselves be verified. The VRS is not a list of verification methods, unless the only requirement regarding each verification activity is that it be performed in a particular way, i.e. a verification solution direction

Acronyms: VRS

Standards: PPA-003914 current release

ARCHITECTURAL DESIGN DESCRIPTION

Description: The Architectural Design Description (ADD) describes the architectural (conceptual) design of the system or subsystem which is the subject of the ADD. The ADD may be supplemented by Interface Design Descriptions (IDDs) and Database Design Descriptions (DBDDs) for descriptions of design decisions relating to external interfaces, internal interfaces, externally input databases, externally output databases and databases internal to the system/subsystem. The ADD, with any associated IDDs and DBDDs, is used to communicate the architectural design within the design team, to design reviewers, acquirers, maintainers and modifiers, as applicable. A System/Subsystem Design Description (SSDD) is a common form of ADD.

Acronyms: ADD, SSDD

Standards: PPA-ME04-002586 current release

INTEGRATED LOGISTICS SUPPORT PLAN – ILSP (1)

1. Prepared by the customer and/or supplier
2. Documents the plan for operational support - it is a support system solution description
3. May include up to ten elements of ILS:
    - Supply Support
    - Support-Related Technical Data
    - Support-Related Facilities
    - Support-Related Manpower and Personnel
    - Packaging, Handling and Storage
    - Operational Training and Training Support
    - Support Equipment
    - Computer Resource Support
    - Maintenance Planning
    - End-Use Item Design Interface.

ILSP (2)

4. Support system requirements must be consistent with readiness or availability requirements or objectives, with End-Use Item design, and with each other

5. Identifies support system structure elements to be developed or acquired so that the End-Use Item is both supportable and supported when released/deployed/installed

6. Includes post-production support to ensure economic logistics support after cessation of relevant production

DETAILED DESIGN DESCRIPTION

Description: A Detailed Design Description (DDD) describes the design of a system or subsystem which is the subject of the DDD, at a level of detail sufficient to allow each element of solution to be acquired or itself be designed and developed. The DDD may be supplemented by Interface Design Descriptions (IDDs) and Database Design Descriptions (DBDDs) for descriptions of design decisions relating to external interfaces, internal interfaces, externally input databases, externally output databases and databases internal to the system/subsystem. The DDD, with any associated IDDs and DBDDs, is used to record and communicate the detailed design within the design team, to design reviewers, acquirers, maintainers and modifiers, as applicable. In practice, a DDD may comprise a plethora of individual design records in a variety of forms

Acronyms: DDD, TDP

Standards:

TEST/VERIFICATION DESCRIPTION

Description: A System or Software Test Description (STD) describes the test/verification preparations, test/verification cases, and test/verification procedures to be used to perform verification testing or other means of verification of a system or system element. The STD is used to communicate to test/verification personnel the information necessary for the test/verification to be performed. The STD also enables the acquirer to assess the adequacy of the verification activity intended to be performed. A STD will normally be prepared in satisfaction of a verification requirement

Acronyms: STD, STP, TD, TP

Standards: TAA-ME04-001136 current release

RECORDS OF TEST/VERIFICATION RESULTS

Description: Test/verification results are the original records of the results of performing verification testing, or other means of verification of a system or system element

Acronyms:

Standards:

VALIDATION PLANS AND PROCEDURES

Description: Validation plans and procedures describe when and how system or system element validation is to be carried out, e.g. test marketing, or operational test and evaluation (OT&E)

Acronyms:

Standards:

RECORDS OF VALIDATION RESULTS

Description: Validation results are the original records of the results of performing validation of a system or system element

Acronyms:

Standards:

OTHER POTENTIAL SYSTEMS ENGINEERING ARTIFACTS

- Feasibility study reports
- Trade-off study reports
- Simulation reports
- Specification tree

FEASIBILITY STUDY REPORTS

Description: A Feasibility Study Report records and communicates the results of a study as to whether is is possible to solve an adequately defined problem, having regard to all the characteristics that must be present in any solution for the solution to be acceptable

Acronyms:

Standards:

TRADE-OFF STUDY REPORTS

Description: A Trade-Off Study Report records and communicates the results of a study as to the overall effectiveness of alternative feasible solutions. A Trade-Off Study Report may also contain the results of optimization of one or more solution alternatives

Acronyms:

Standards:

SIMULATION REPORT

Description: A Simulation Report records and communicates the results of the conduct of simulation activities. Simulation activities are activities which seek to imitate the behavior of something by means of the behavior some other thing suitably analogous

Acronyms:

Standards:

SPECIFICATION TREE

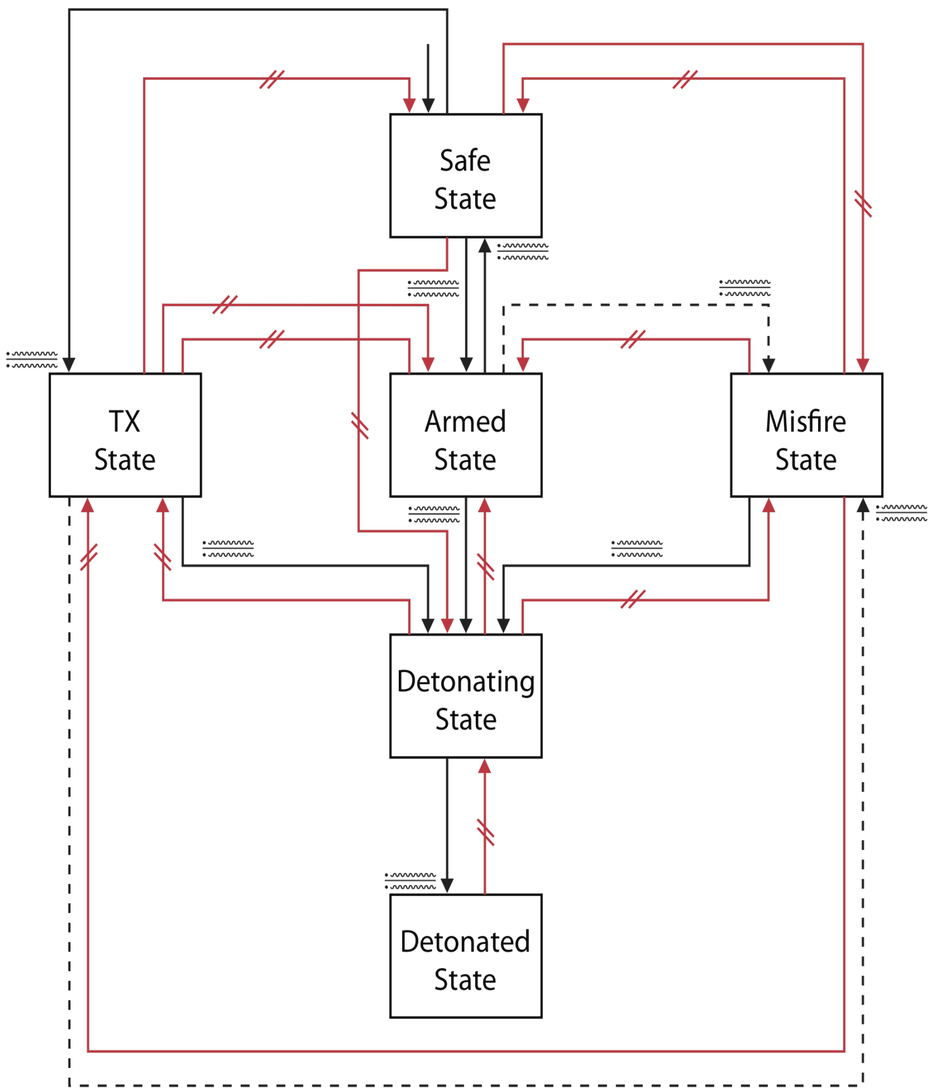
Description: A specification tree shows the requirements specifications related to a technical system under development in a hierarchical order related to the structure of the system in terms of system elements

Acronyms:

Standards:

MODEL-BASED SYSTEMS ENGINEERING (MBSE) APPLIED TO THE PROBLEM DOMAIN

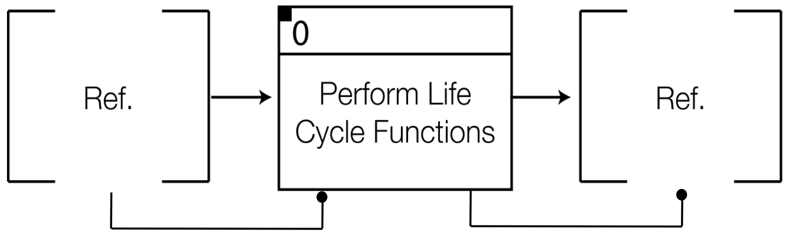
STATES & MODES ANALYSIS EXAMPLE



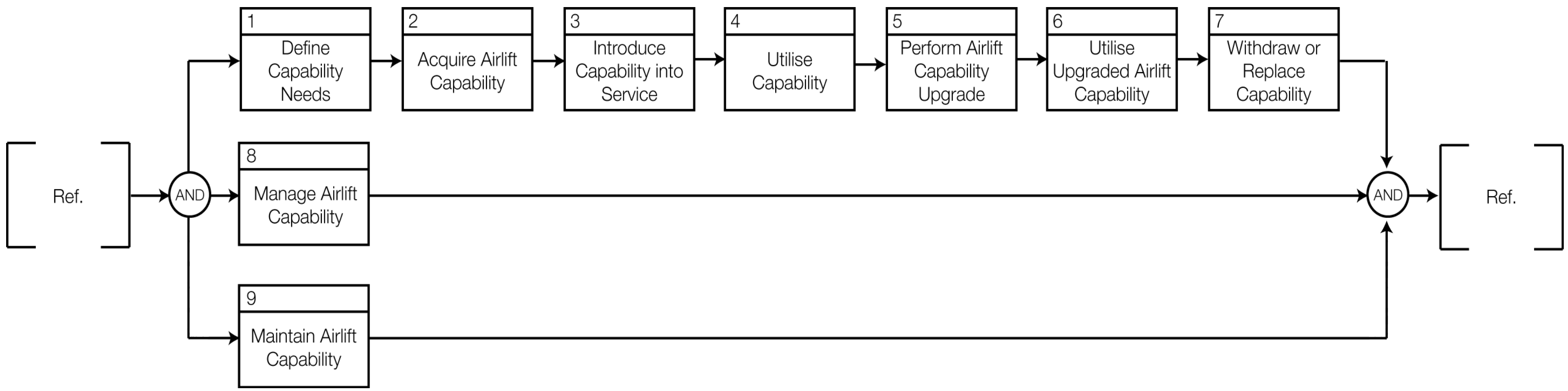
Legend:

TX Time Expired
Required transition
Permitted but not required transition
Prohibited transition
Unconnected arrow : Default state or mode

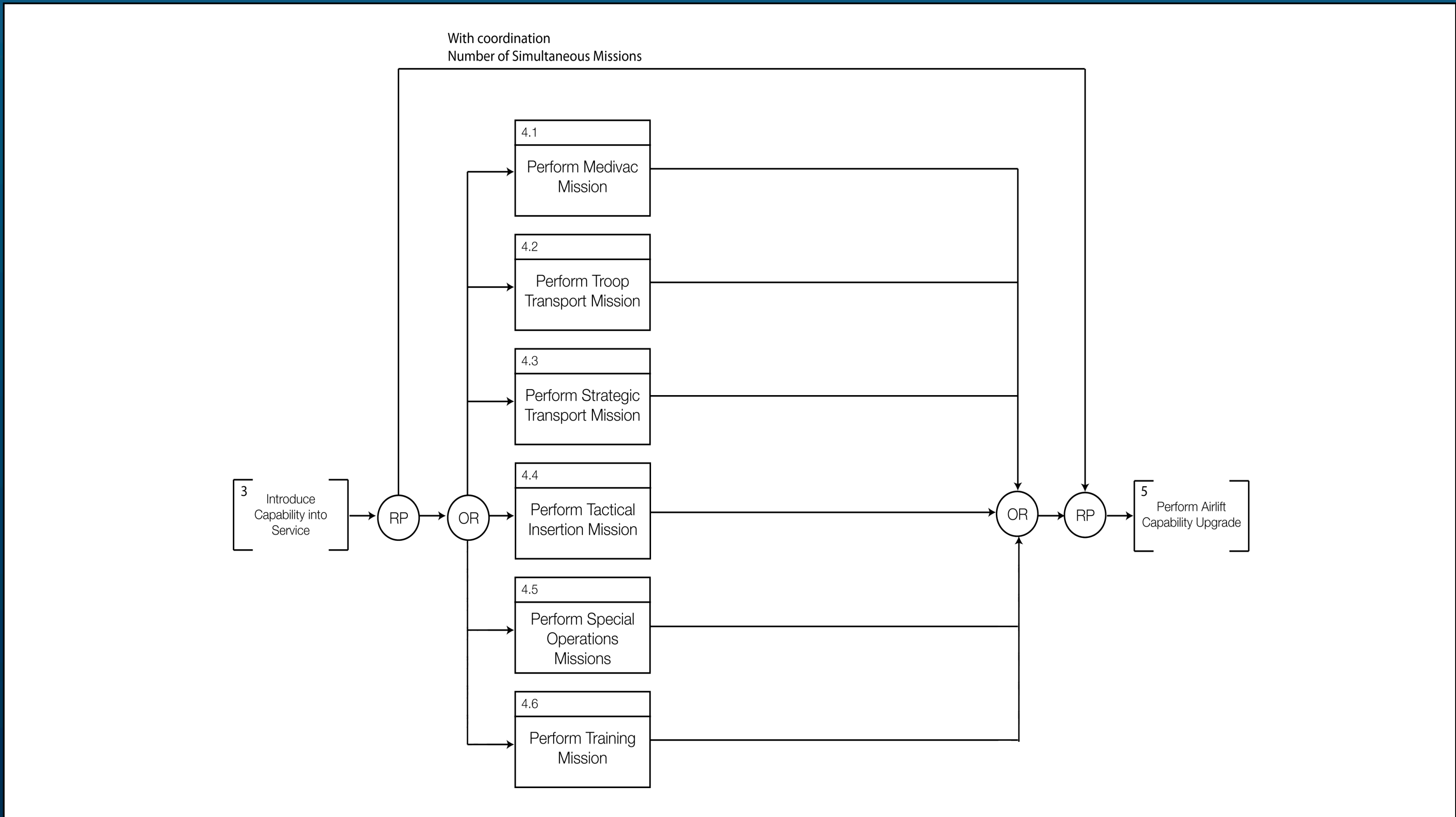
FUNCTIONAL ANALYSIS - EXAMPLE 1-1



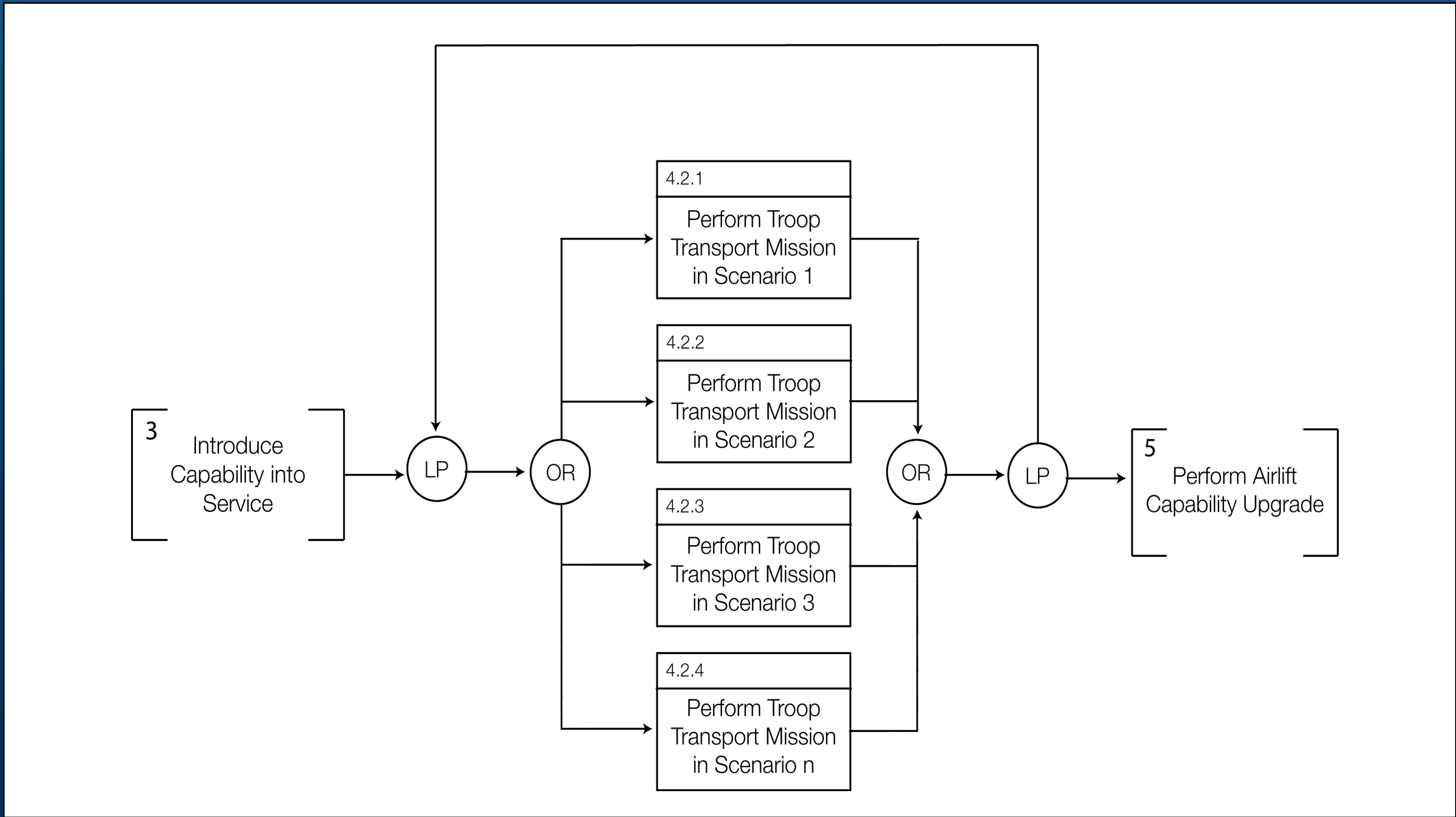
FUNCTIONAL ANALYSIS - EXAMPLE 1-2



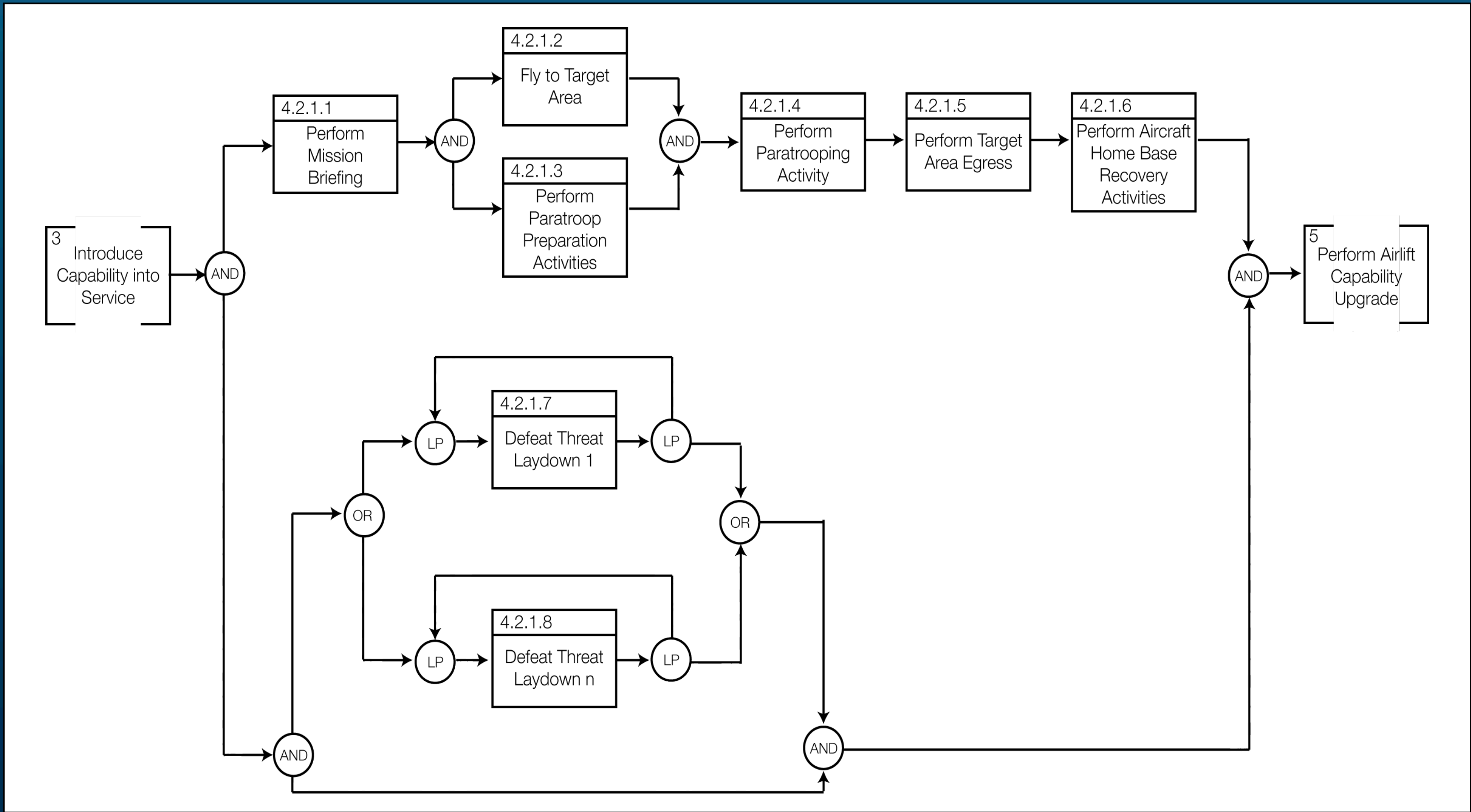
FUNCTIONAL ANALYSIS - EXAMPLE 1-3



FUNCTIONAL ANALYSIS - EXAMPLE 1-4

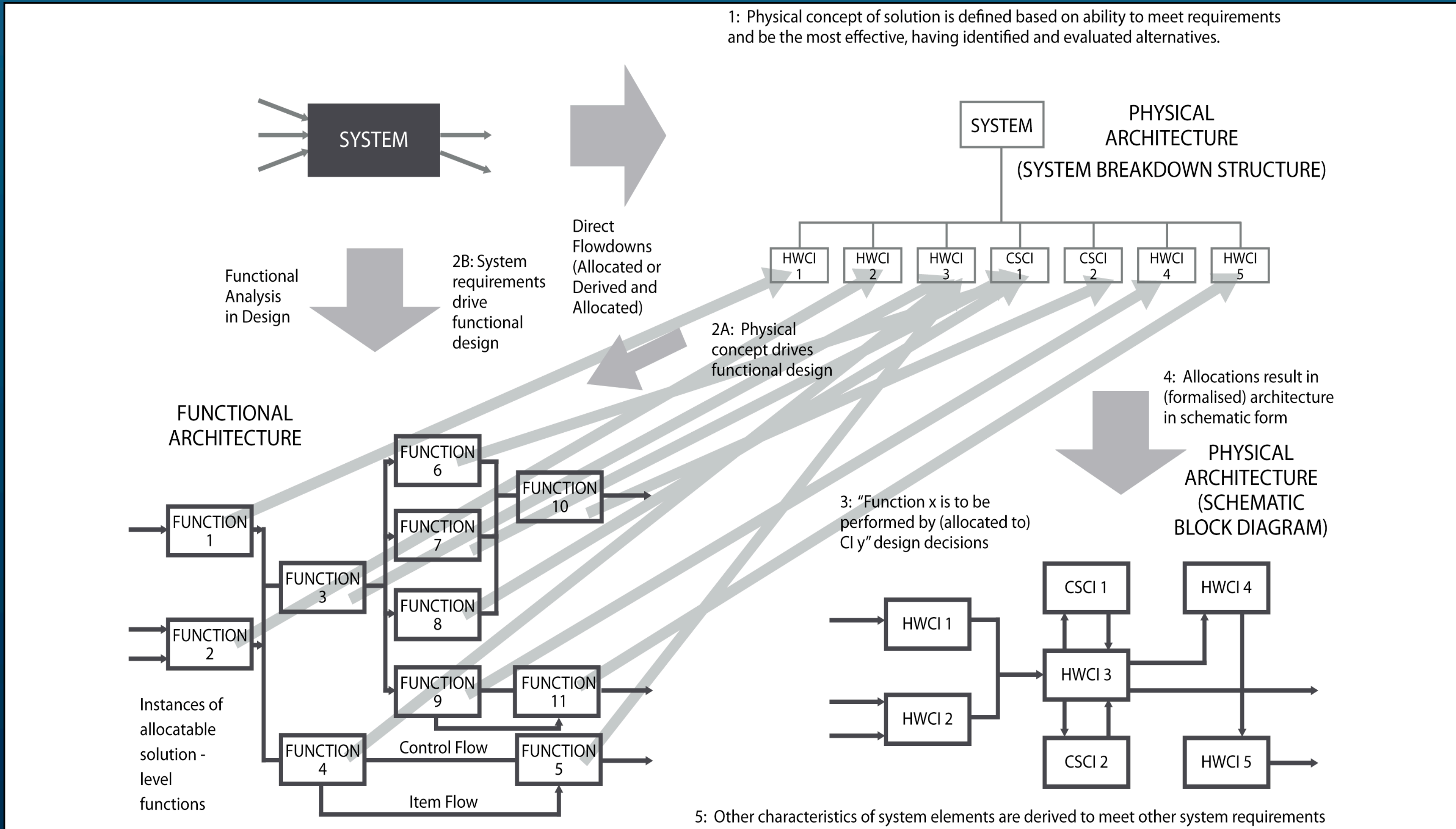


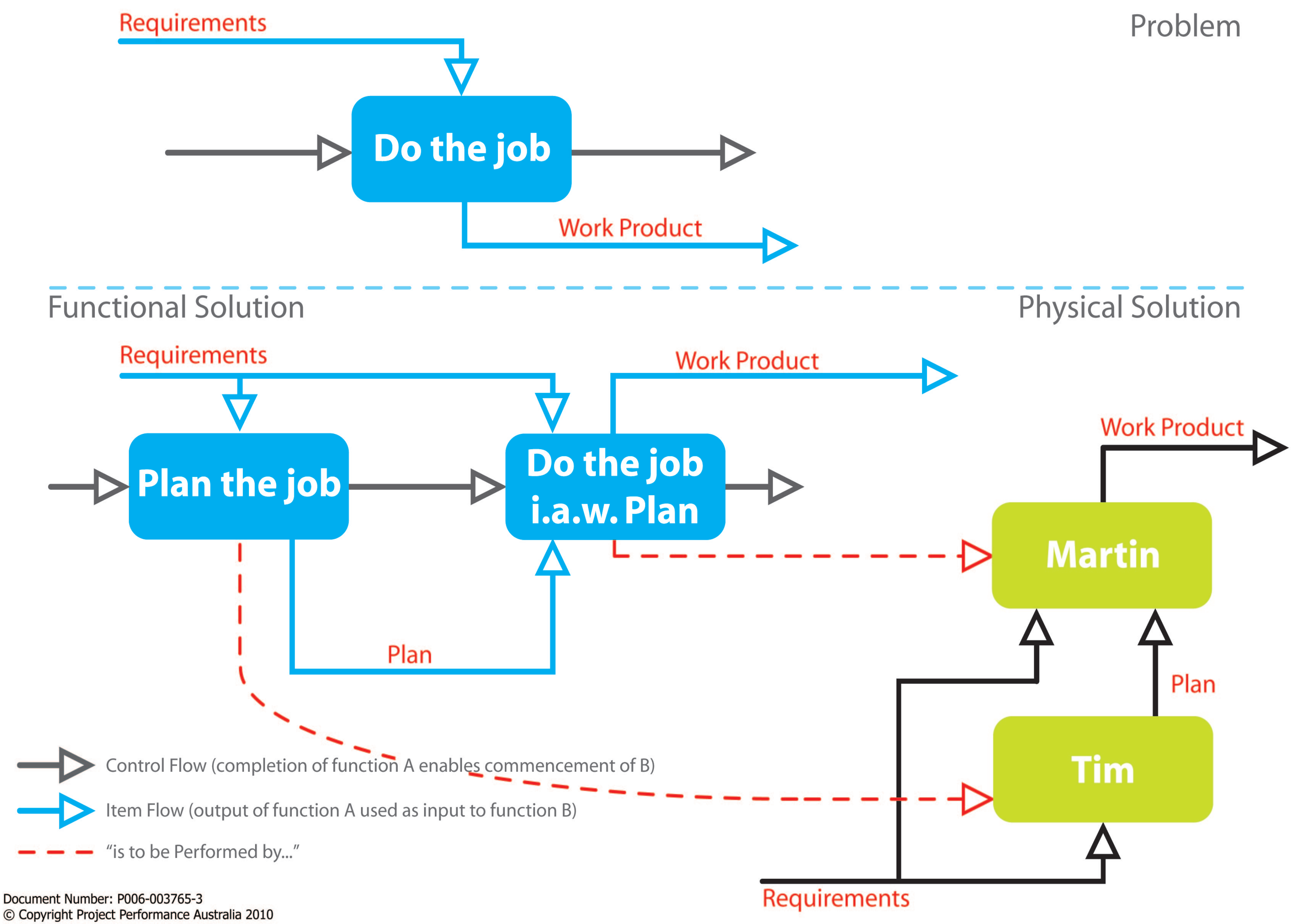
FUNCTIONAL ANALYSIS - EXAMPLE 1-5

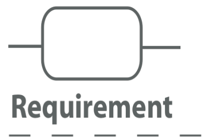


MODEL-BASED SYSTEMS ENGINEERING (MBSE) APPLIED TO THE SOLUTION DOMAIN

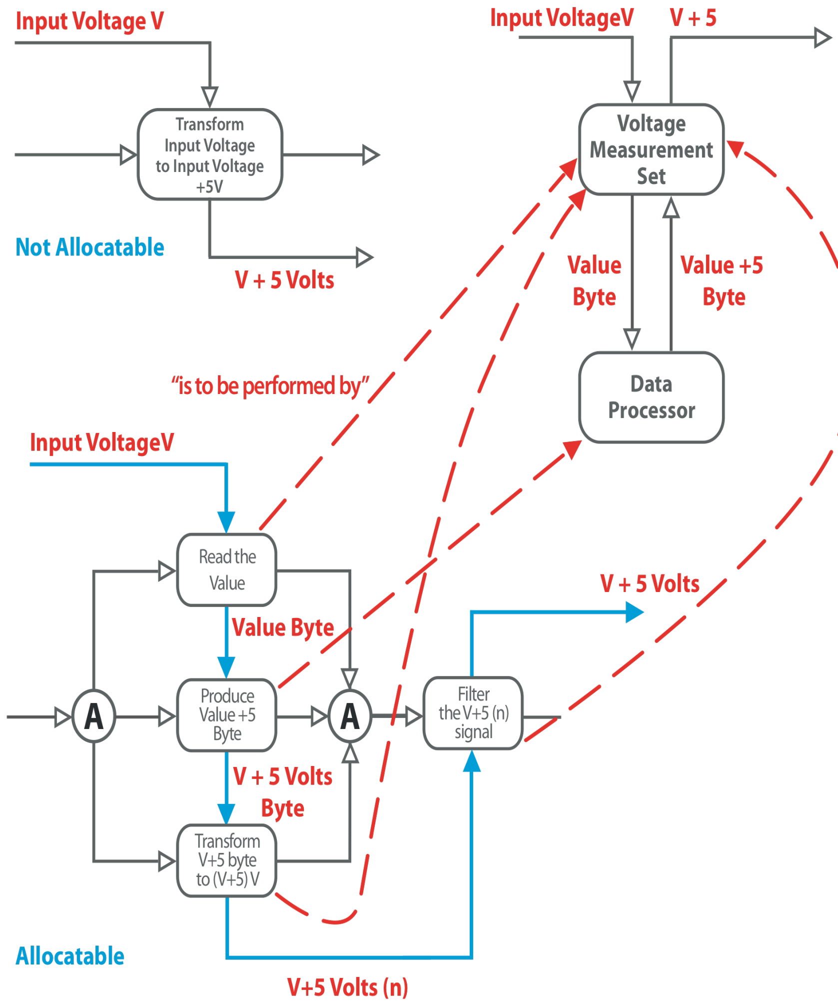
ALLOCATION OF FUNCTIONS TO ARCHITECTURAL ELEMENTS



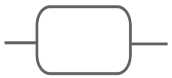

Problem
Requirements
Do the job
Work Product
Functional Solution
Physical Solution
Requirements
Work Product
Plan the job
Do the job i.a.w. Plan
Plan
Work Product
Martin
Plan
Tim
Requirements
Control Flow (completion of function A enables commencement of B)
Item Flow (output of function A used as input to function B)
"is to be Performed by..."



(Incomplete) Functional Design

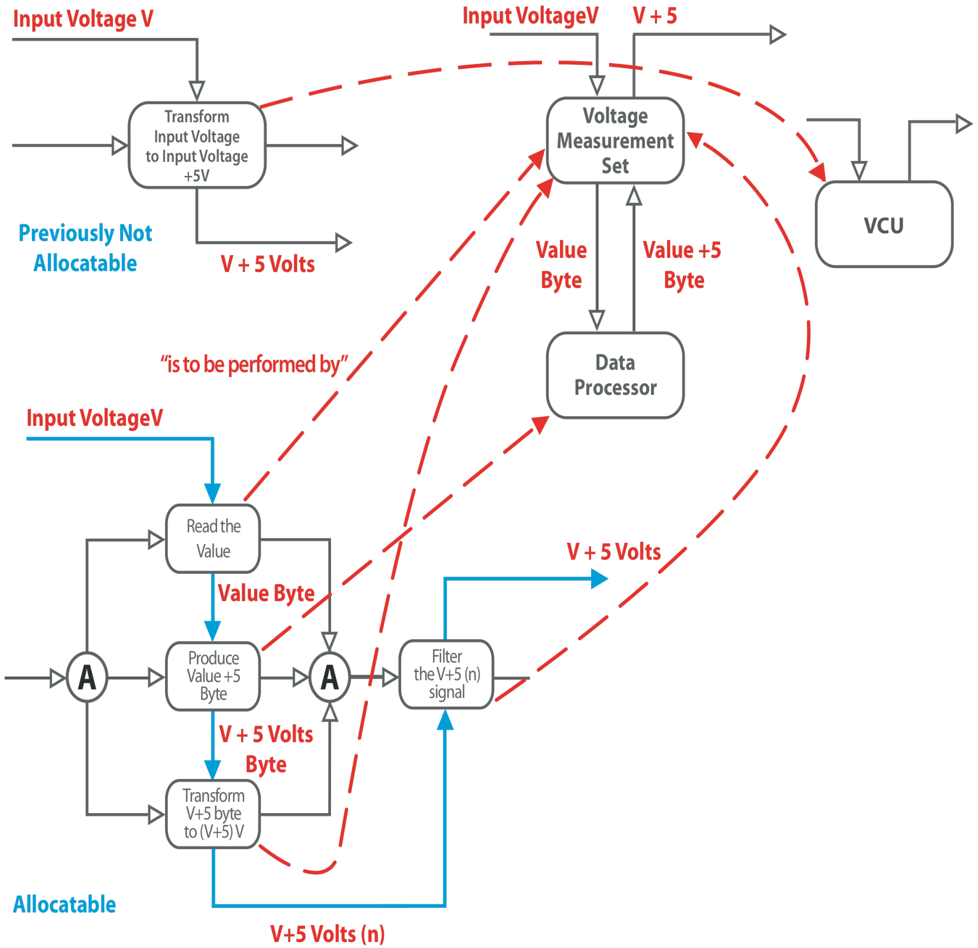


Document Number: P006-003763-3

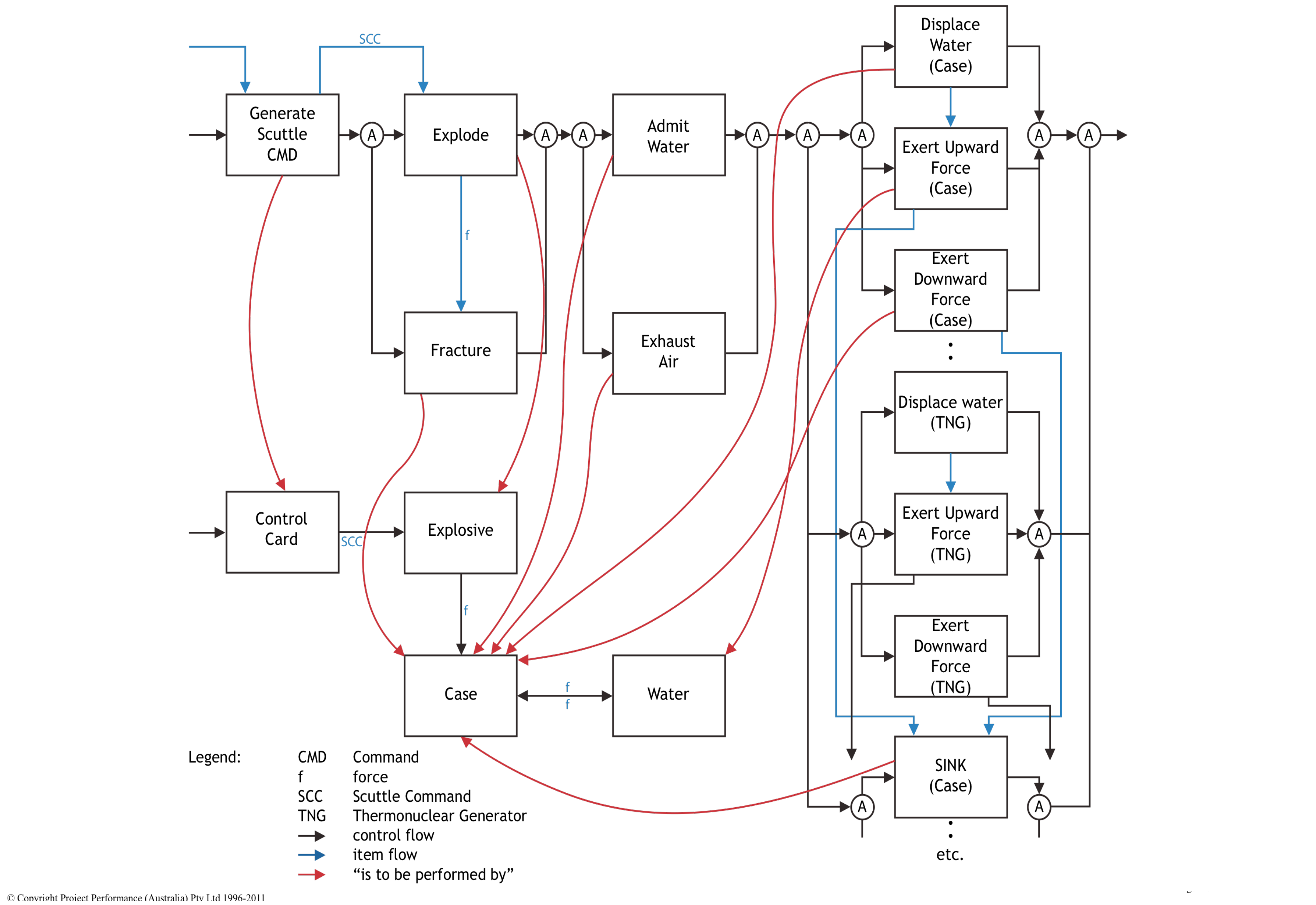


Requirement

Functional Design

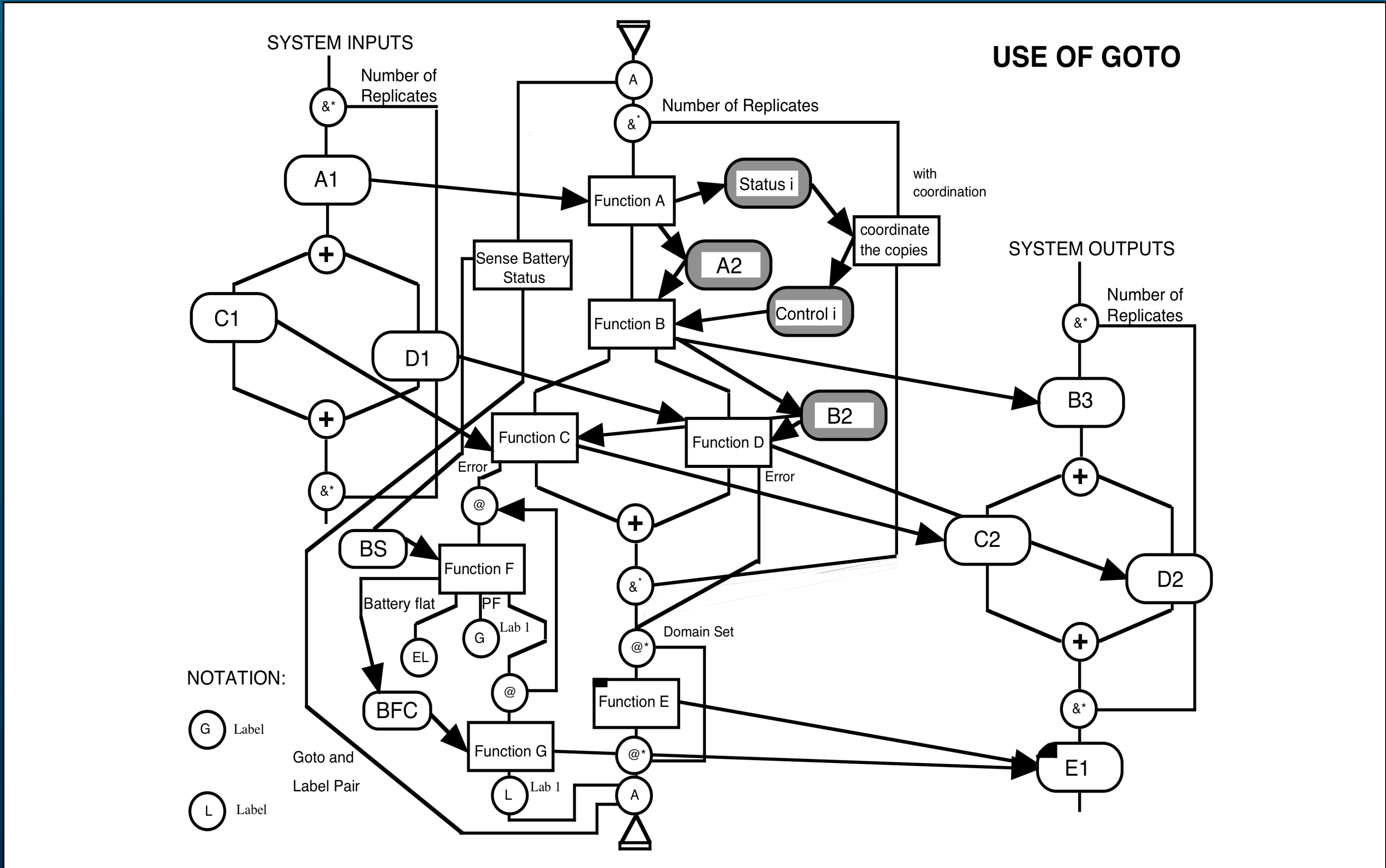


Document Number: P006-004747-2

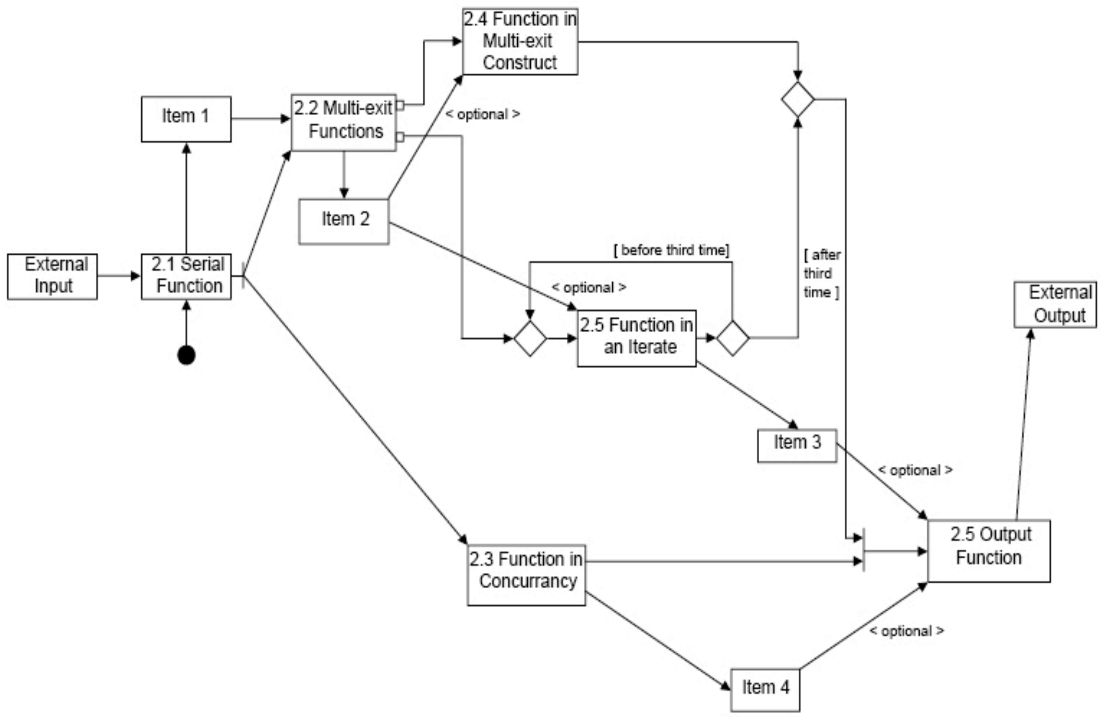


SCC
Generate Scuttle CMD
A
Explode
A
A
Admit Water
A
A
A
Displace Water (Case)
Exert Upward Force (Case)
A
A
f
Fracture
Exhaust Air
Exert Downward Force (Case)
Displace water (TNG)
Control Card
SCC
Explosive
A
Exert Upward Force (TNG)
A
f
Exert Downward Force (TNG)
Case
f
f
Water
SINK (Case)
A
A
etc.
Legend:
CMD Command
f force
SCC Scuttle Command
TNG Thermonuclear Generator
control flow
item flow
"is to be performed by"
© Copyright Project Performance (Australia) Pty Ltd 1996-2011

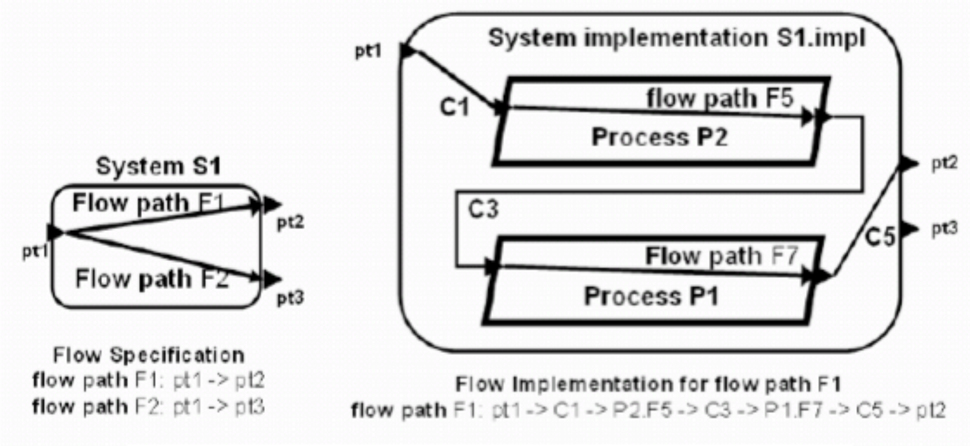
RDD BEHAVIOUR MODEL



SYSML - FUNCTIONAL MODELING



AADL DIAGRAM



A graphical representation

```
/osamplesteps

eriodic;

    .outed -> compute1.ined;
  compute12: data port compute1.outed ->> compute2.ined;
  compute23: data port compute2.outed -> compute3.ined;
  actuateconn: data port compute3.outed ->> actuate.ined;
  bus access db -> sense.devbus;
  bus access db -> actuate.devbus;
flows
  etelatency: end to end flow sense.flow1 -> senseconn -> com-
pute1.flow1
           -> compute12 -> compute2.flow1 -> compute23 -> com-
pute3.flow1
           -> actuateconn -> actuate.flow1 { latency => 153 ms;};
end application.twosamplesteps;
```

A textual representation

SOFTWARE SUPPORT TO SYSTEMS ENGINEERING

- Requirements management software
- Value modeling and decision support software
- Logical and physical design software
- Design analysis software
- Simulation software
- Test management software
- Configuration management software

HARDWARE SUPPORT TO SYSTEMS ENGINEERING

- Computing hardware and peripherals
- Test equipment
- Physical prototypes

A Project Performance International parabeniza os membros do INCOSE Brazil e deseja todo o sucesso ao Capitulo

Robert J. Halligan
Email: rhalligan@ppi-int.com