Mission Definition and Design

Space Systems Engineering

A. Castro

BCN Technoweek
June 2019

Introduction

- Space projects are:
  - Complex and Demanding
  - Technically Novel
  - Deployed in Harsh Environment
  - Operate Remotely
  - Financially & Time Constrained

Image credit: ESA

- Therefore => high technical risk
- Demands application of sound Systems Engineering practices

Introduction

- Systems Engineering provides:
  - Inter-disciplinary approach
  - Effective management of complexity
  - Management of technical maturity risk
  - Methodologies for the management of:
    - Schedule
    - Cost
    - Technical performance
    - Risks

Introduction

- Some space systems engineering features:
  - Space systems engineering plays different role in the different phases of the project life cycle.
  - Interacts with technical management/project management.
  - It has specific processes and methodologies.
  - It is interdisciplinary and links technical, risk, programmatic and cost.
  - Defines the interactions and inter-dependencies of the different system functionalities.
  - Address interfaces between subsystems/segments, as well as external ones.
  - In line with ECSS standards (and other standards).

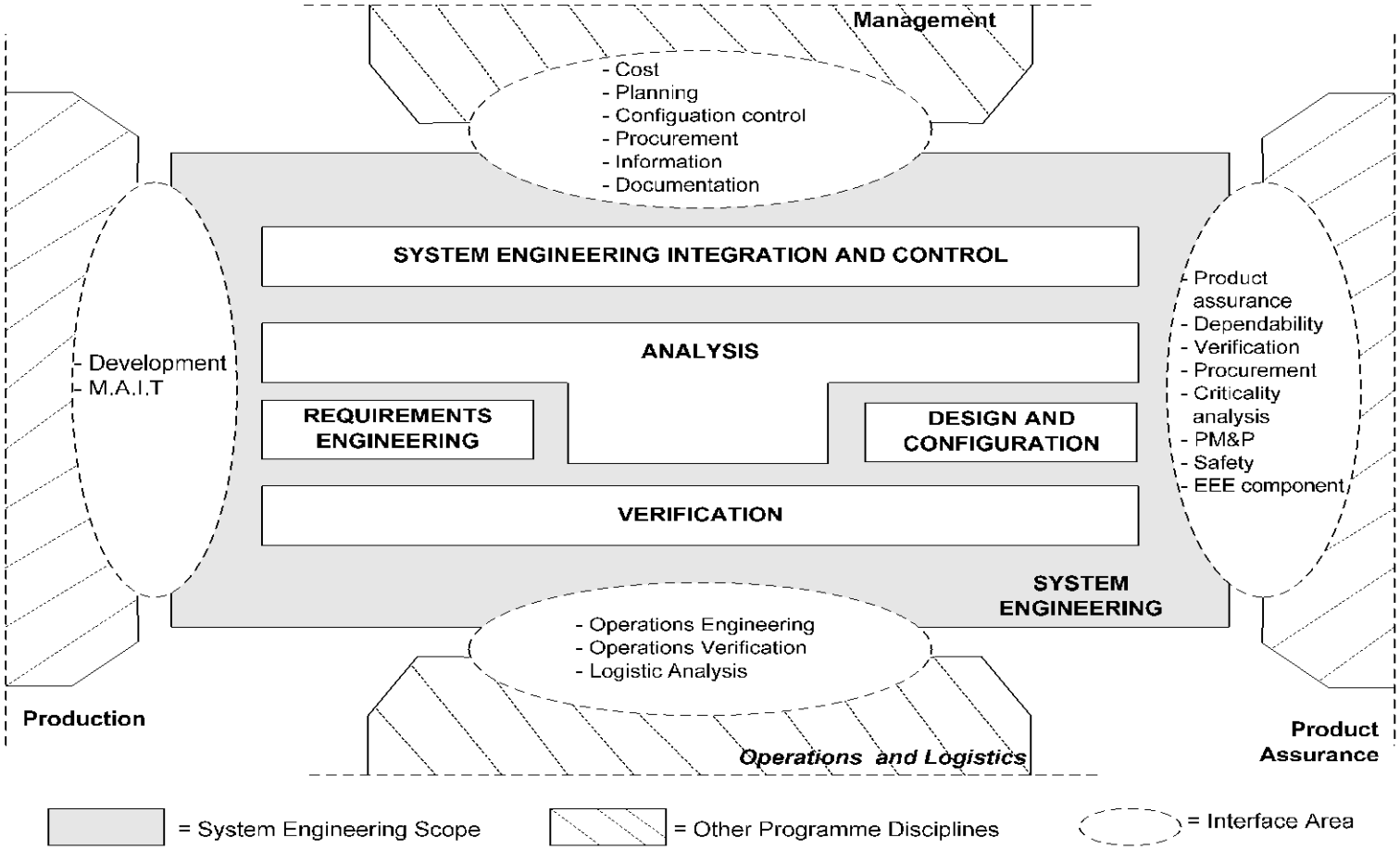
Systems Engineering Definitions

- *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, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. Systems engineering integrates all 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 business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.* (INCOSE, 2004)

- System
  << *A set of functional elements organized to satisfy user needs* >> (IEEE P1220)
- Mission
  << *Specific task, duty or function defined to be accomplished by a system* >> (EN 13701:2001)
- Requirement
  << *Need or expectation that is stated, generally implied or obligatory* >> [ISO 9000:2000]

- Formal definition
  << *The interdisciplinary approach governing the total technical effort required to transform a requirement into a system solution* >> (ECSS E-ST-10C)
- Practical definition
  The "technical management" of a product design and development

ESA Systems Engineering Functions



MAIT = manufacturing, assembly, integration and test
PM&P = parts, materials and processes

ESA Systems Engineering Functions

Space System

Space Segment

Spacecraft/Platform

The system composed of one or more elements that carries and provides all the required services to the payload

Payload

The set of instruments that achieve the (science) objectives of the mission

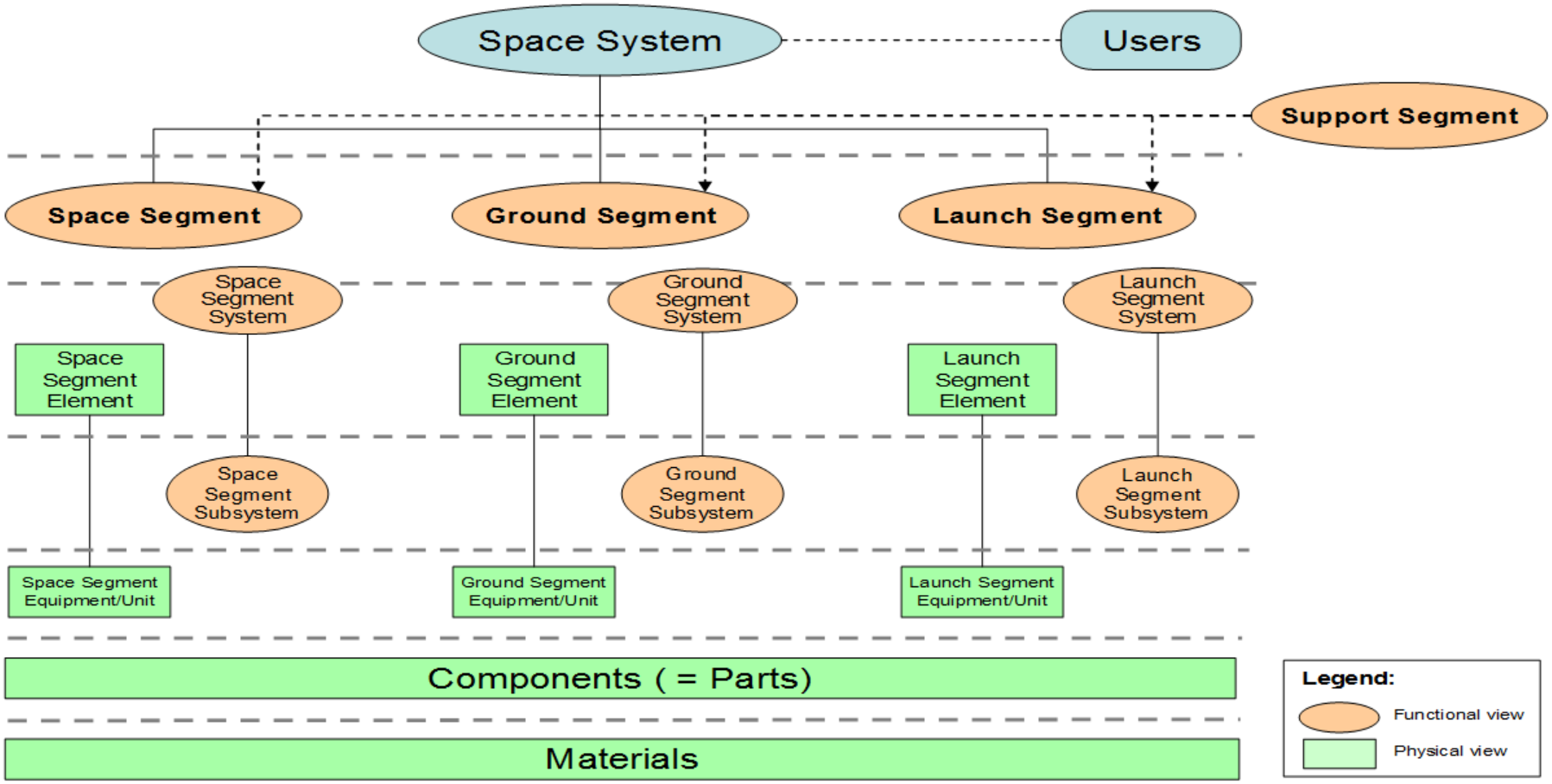
Ground Segment (incl. Operations)

The set of activities to operate the spacecraft and the payload from Earth

Launcher

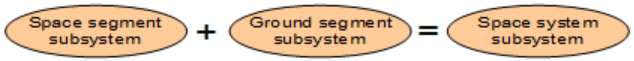
The system design shall find the optimal combination of the elements above to achieve the given mission objectives

Space Systems Breakdown



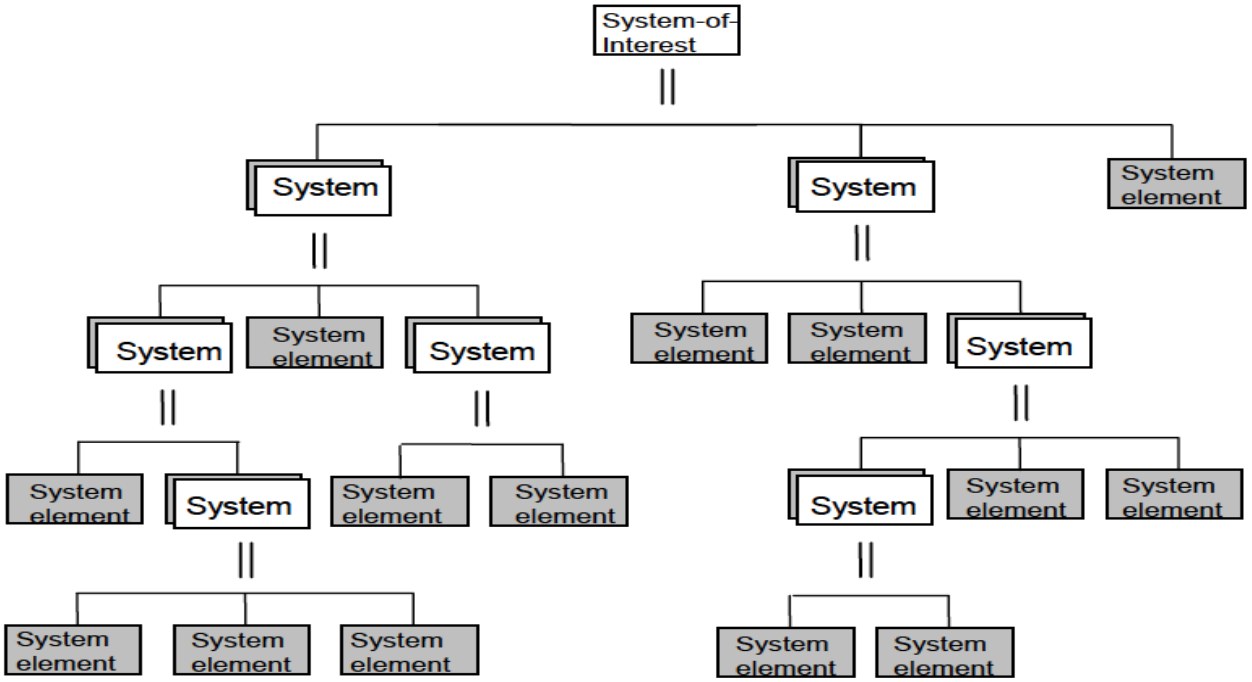
Note 1: Since software can belong to any level it is not apparent in this chart

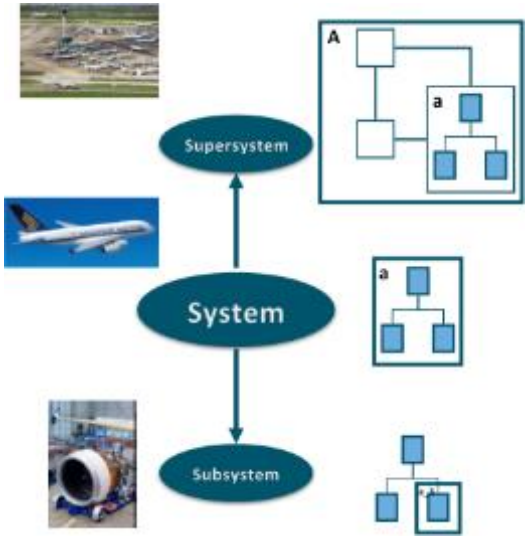
Note 2: A subsystem can be split across two segments
e.g. TT&C subsystem split across Space and Ground segments



Hierarchy of System, System Element and System of Interest

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





System Development Lifecycle

1. Conceptual design and requirements definition

2. Design

3. Qualification (*verification that the system design fulfils the specified requirements with a margin)*

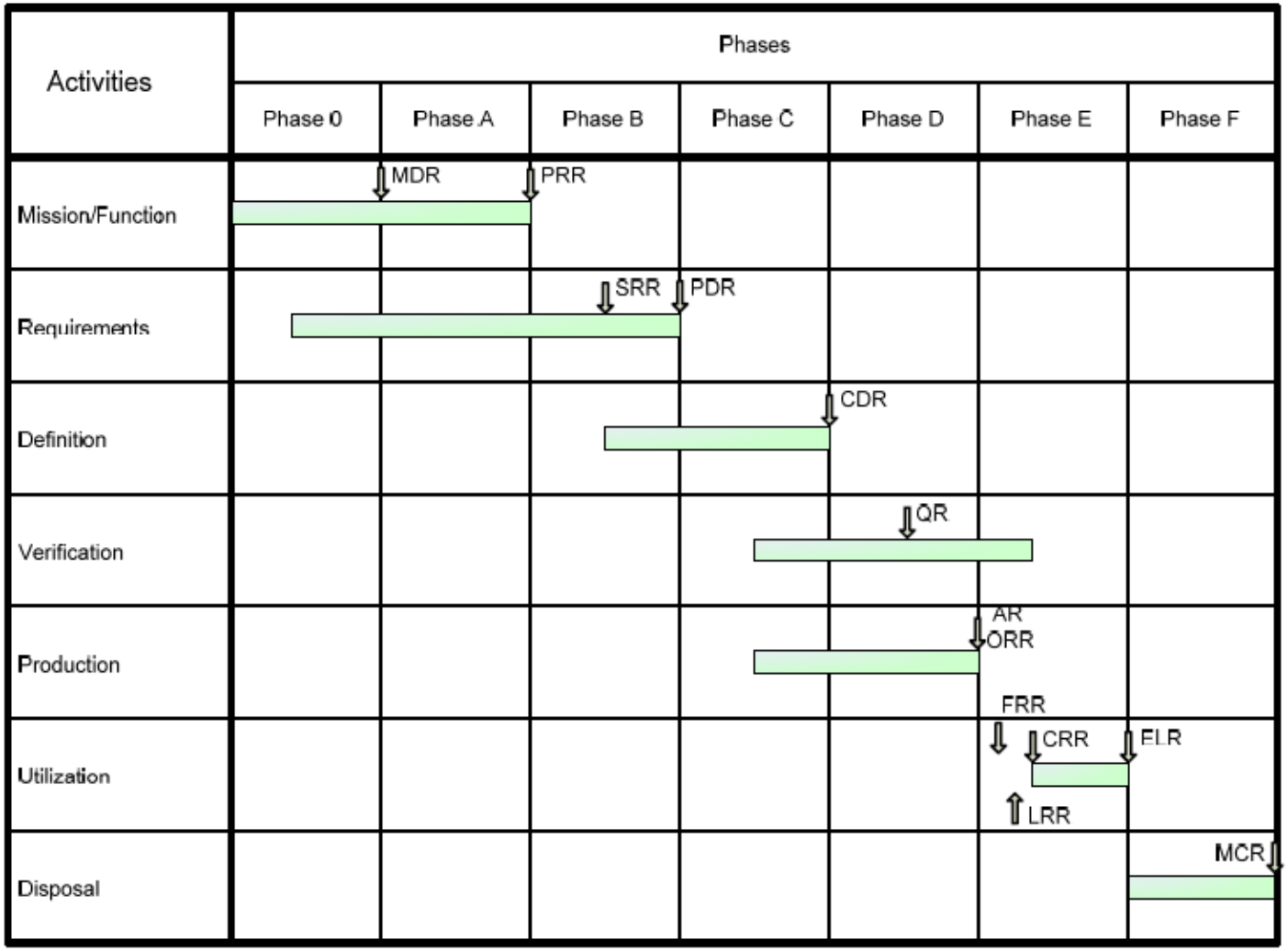
4. Production

5. Customer Acceptance (*check that the product is in agreement with the qualified design, is free from workmanship defects and acceptable for use)*

6. Utilisation/Operation

7.Disposal

Space Mission Phases



ECSS-M-ST-10C Figure 4-3

Space Mission Phases

Phase	Definition	Duration	Main Events
Phase 0	Identification of mission requirements	1-2 months	Issue of Mission Requirement Document
Phase A	Feasibility	1 year	Issue of Mission Assessment Report
Phase B	Preliminary Design	1 year	Freezing of System Requirements (System Requirement Review) Key design checkpoint (PDR)
Phase C	Detailed Design	6 months-1 year	Critical Design Review
Phase D	Qualification and Production	3-7 years	Models Testing, Qualification Review, Hardware building and Assembly, Flight Acceptance Review
Phase E	Launch and Utilisation	Many years	Launch, Spacecraft Commissioning, Nominal Operations, Contingencies
Phase F	Disposal	Few days	De-orbit

Review objectives

Phase	Review	What are we trying to establish?
0	Mission Definition Review MDR	Do we understand and have we expressed what this mission is expected to do?
A	Preliminary Requirements PRR	Have we been able to express a set of top level requirements and plans, and do these show technical and programmatic feasibility?
B	System Requirements Review SRR	Are the requirements and understanding of the system design and verification approach now maturing to the extent we expect?
B	Preliminary Design Review PDR	Have we reached a level of preliminary design that we are confident to progress to detailed design, including the technical and management planning?
C	Critical Design Review CDR	Are we confident that we are ready to start manufacture, assembly, integration and test?
D	Qualification Review QR	Does the design, including margins, meet the requirements and do we have the evidence to prove this?
D	Acceptance Review AR	Is the 'product', including all necessary information, ready to transition to operations and do we have the evidence to prove this?
D	Operational Readiness Review ORR	Do we have the necessary and correct operational procedures and teams in place. Is the ground segment ready for operations?

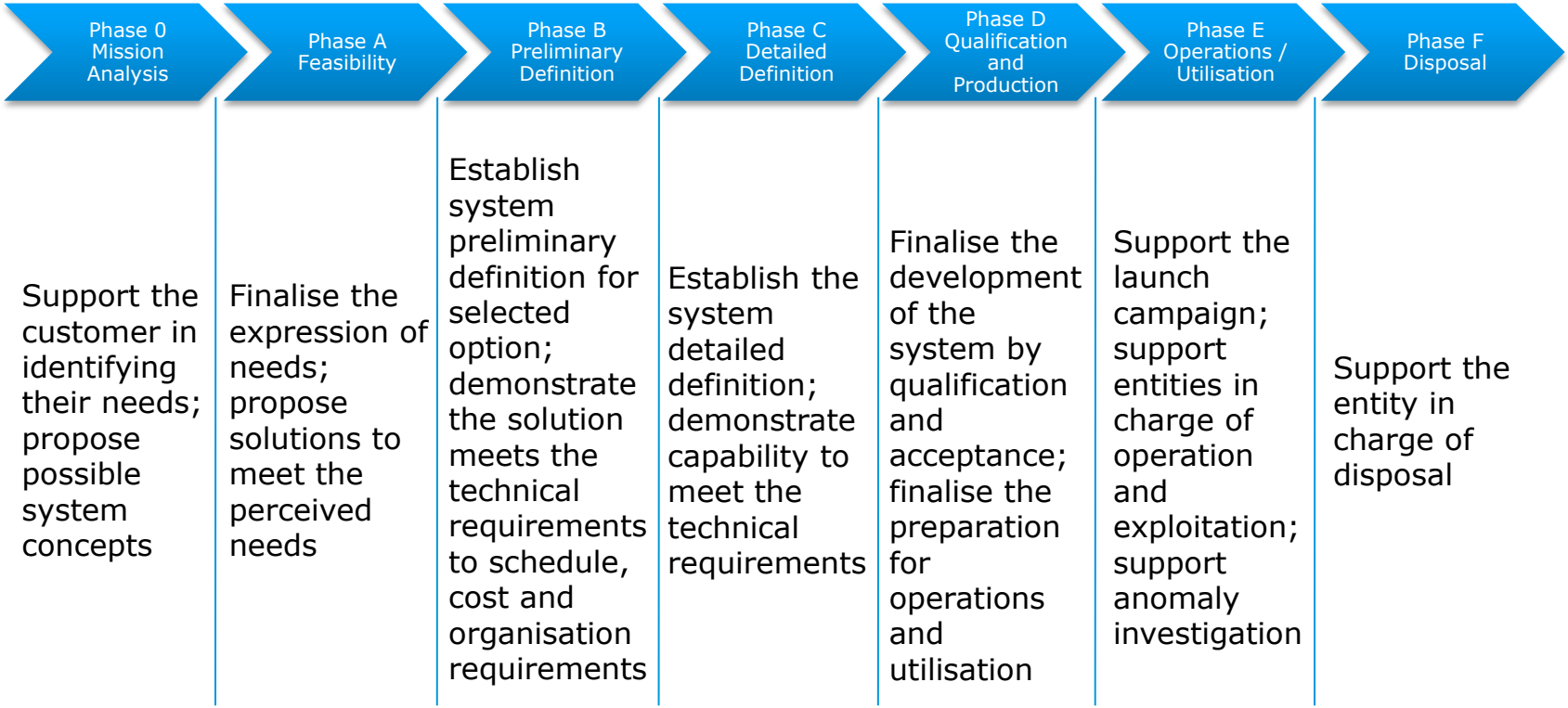
ECSS-M-ST-10C Section 4.3

Review objectives

Phase	Review	What are we trying to establish?
E	Flight Readiness Review FRR	Are all of the necessary flight and ground systems ready for launch?
E	Launch Readiness Review LRR	Declaration of readiness for launch to provide the authorisation to proceed for launch
E	Commissioning Results Review CRR	Can we provide evidence that all elements of the system are performing within the specified parameters? Can we declare readiness for routine operations / utilisation?
E	End of Life Review ELR	Has the mission completed its useful operations? Can we ensure that the space segment is configured for safe disposal?
F	Mission Close-out Review MCR	Have we satisfactorily completed all mission disposal activities?

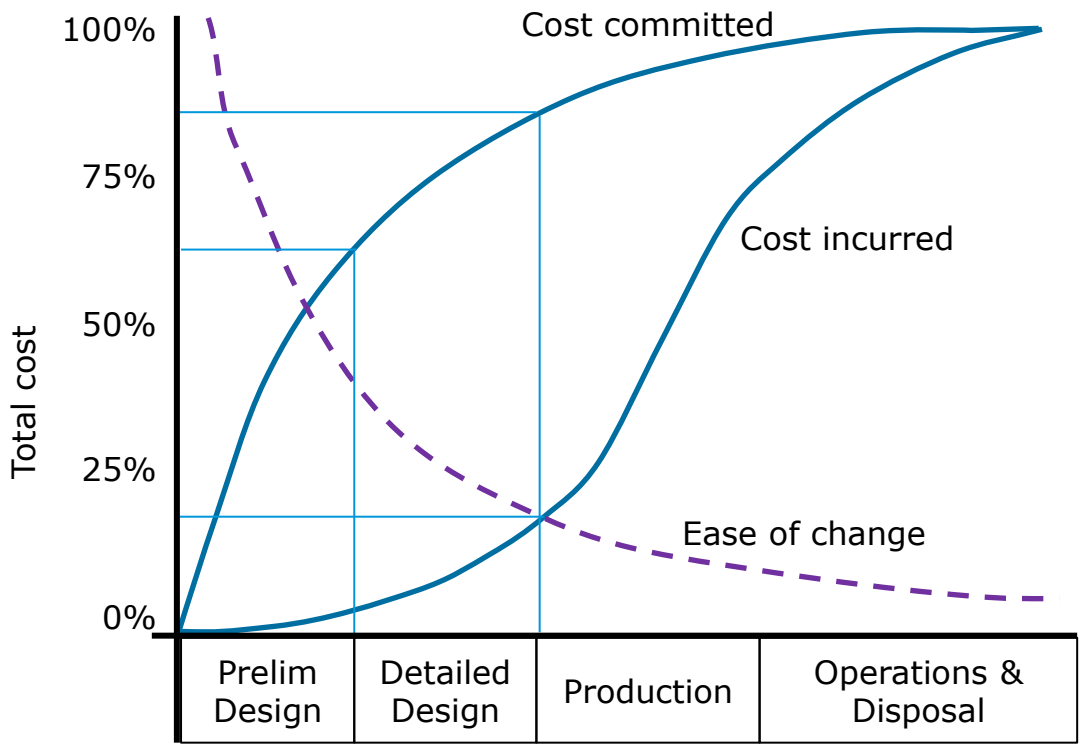
ECSS-M-ST-10C Section 4.3

Mission phases & lifecycle



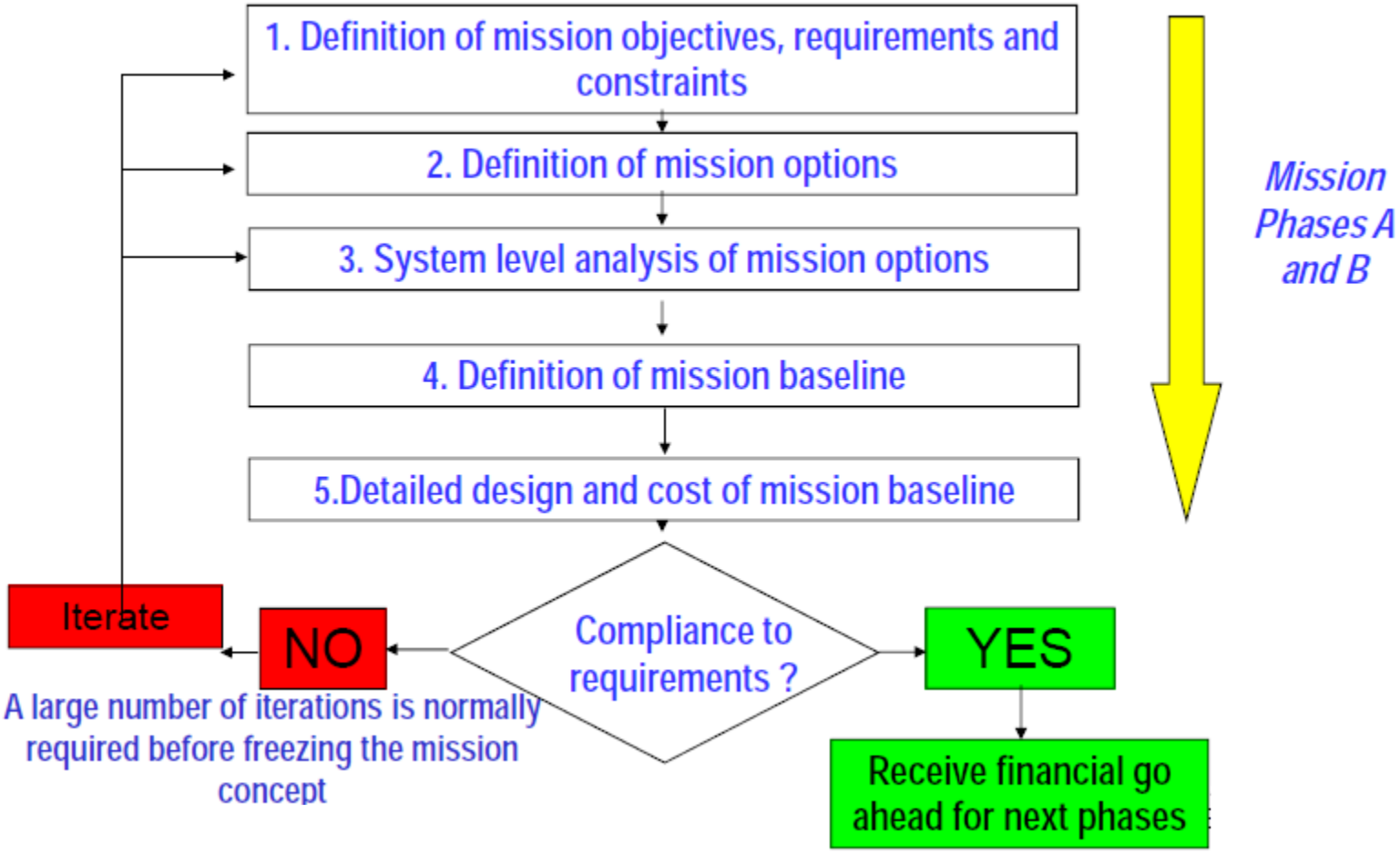
ECSS-E-ST-10C Section 6

Decisions Across The Lifecycle

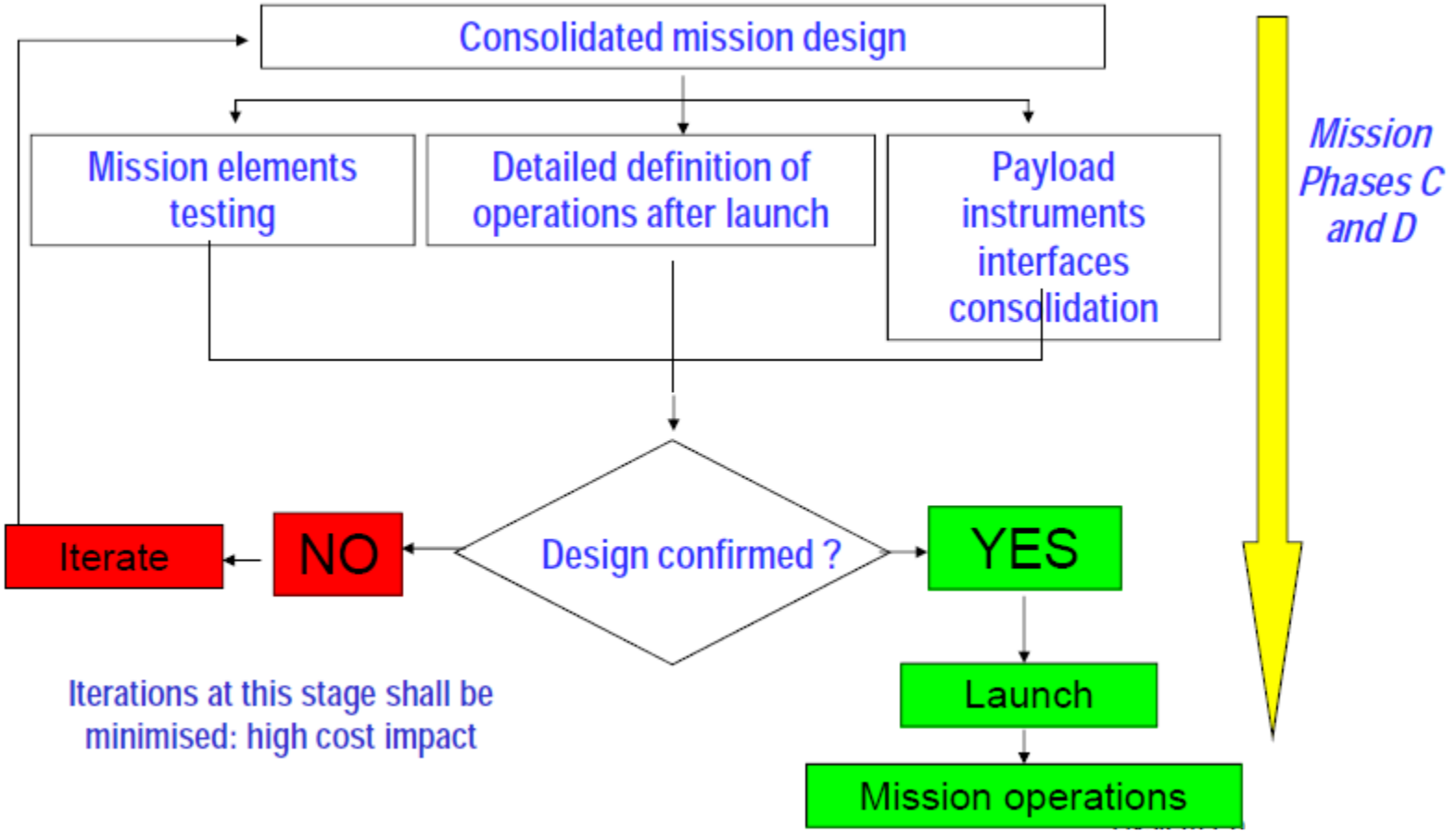


- Early decisions set the committed cost
- Problems are cheaper to resolve early on

Space Mission Steps and Iterations



Space Mission Steps and Iterations



Activities in the Early Phases of a Project (phase 0 and phase A)

1. Translation of customer needs and requirements in mission/system technical requirements
2. Definition of mission/system options in compliance with the technical requirements
3. Interface to mission analysis
4. Options trade and definition of mission baseline
5. Overall design consistency
6. System harmonisation and optimisation (sensitivity analyses)
7. System budgets and design margins management
8. Preliminary definition of interfaces (launcher, payload)

Activities in the Design Phases of a Project (phase A to early C)

- Flow-down of requirements to subsystems
- Interface to mission analysis
- Overall design consistency and budgets
- Management of design margins
- System harmonisation and optimisation
- Definition of development process
- Support to operations definition
- Management of interfaces

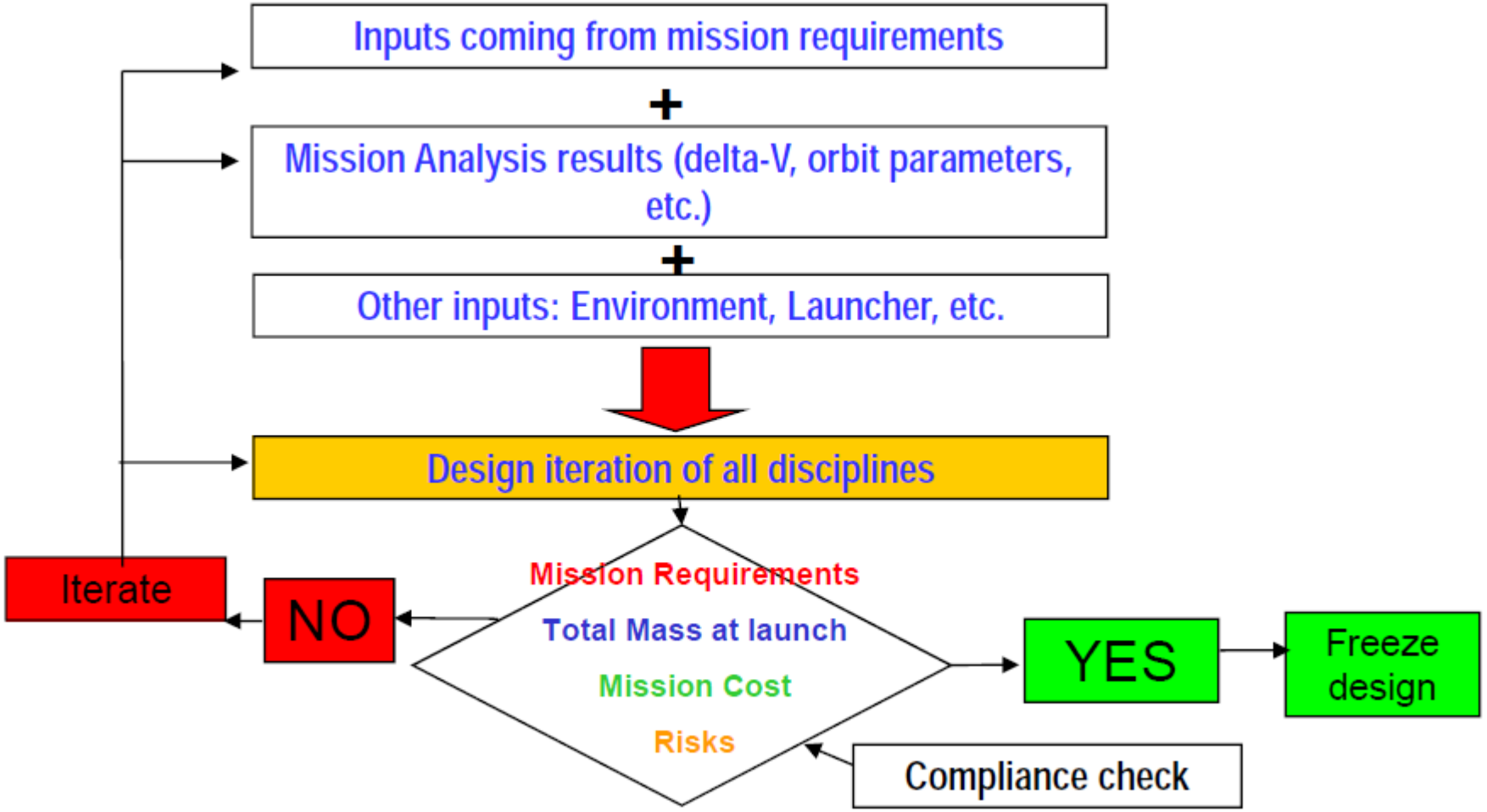
Development Phases of a Project (phase C to phase E)

- Traceability of requirements
- Support to AIV and Product Assurance in procurement, assembly and verification
- Overall design consistency and budgets
- Management of design margins
- Support to operations definition
- Management of interfaces

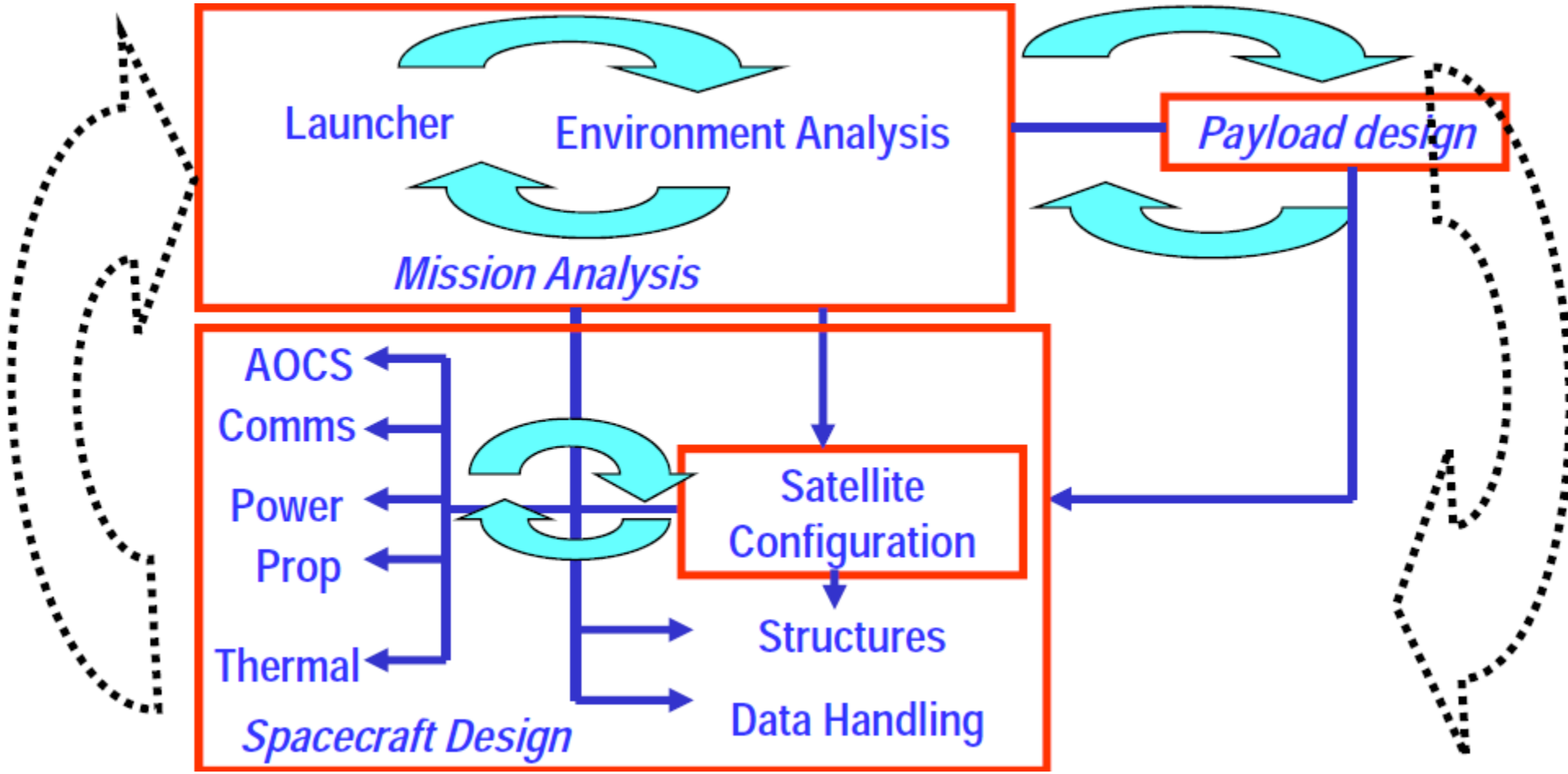
Design Flow

- The design of all subsystems is highly iterative
- It is fundamental to define properly a design flow that the system engineer shall initiate with first inputs to the other disciplines and shall control by means of defined parameters (e.g. mass)
- The system engineer shall define when the design has reached a sufficient level of maturity and no further iterations are needed

Space Mission Design Flow



Space Mission Design Flow



Design Iterations

- Many design iterations of the different spacecraft subsystems are needed to converge to a solution that meets requirements.
  Causes:
  - Resources (mass, volume, power, data, etc.) are always at a premium and subsystems "fight" each other for a share of them
  - Requirements are often conflicting: compromises shall be sought
  - "External" (e.g. political) inputs often become available in the middle of the design as they are not synchronous with the technical work

Space Mission Analysis & Design

Platform | Payload

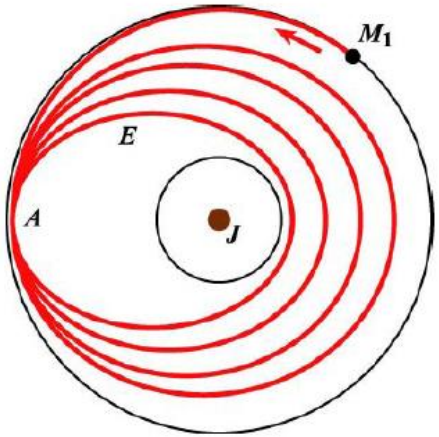
Space Segment

Mission Definition

Mission Exploitation

Mission Operations

Orbit



Launcher

Mission Architecture Patterns

	Earth Observation	Remote science	Planetary observation
Orbit	Mostly LEO	Any	Long duration transfer, gravity assist, planet capture
Space Segment	Spacecraft families, Prime contractor model	Typically bespoke, consortium model	Often bespoke → s/c families
Launcher	Vega, Rockot, Soyuz	Vega, Soyuz or partner launcher	Soyuz, Ariane 5 or partner launcher
Mission Operations	ESOC, Eumetsat	ESOC or partner space agency	ESOC or partner space agency
Mission Exploitation	ESRIN, Eumetsat + value-added industry	Science consortium, PIs, National space agencies	Science consortium, PIs, National space agencies

Limited cases for illustrative purposes. Information is partial

Translating needs to specifications and plans

Mission Need

Mission Analysis And Design

Mission Pattern

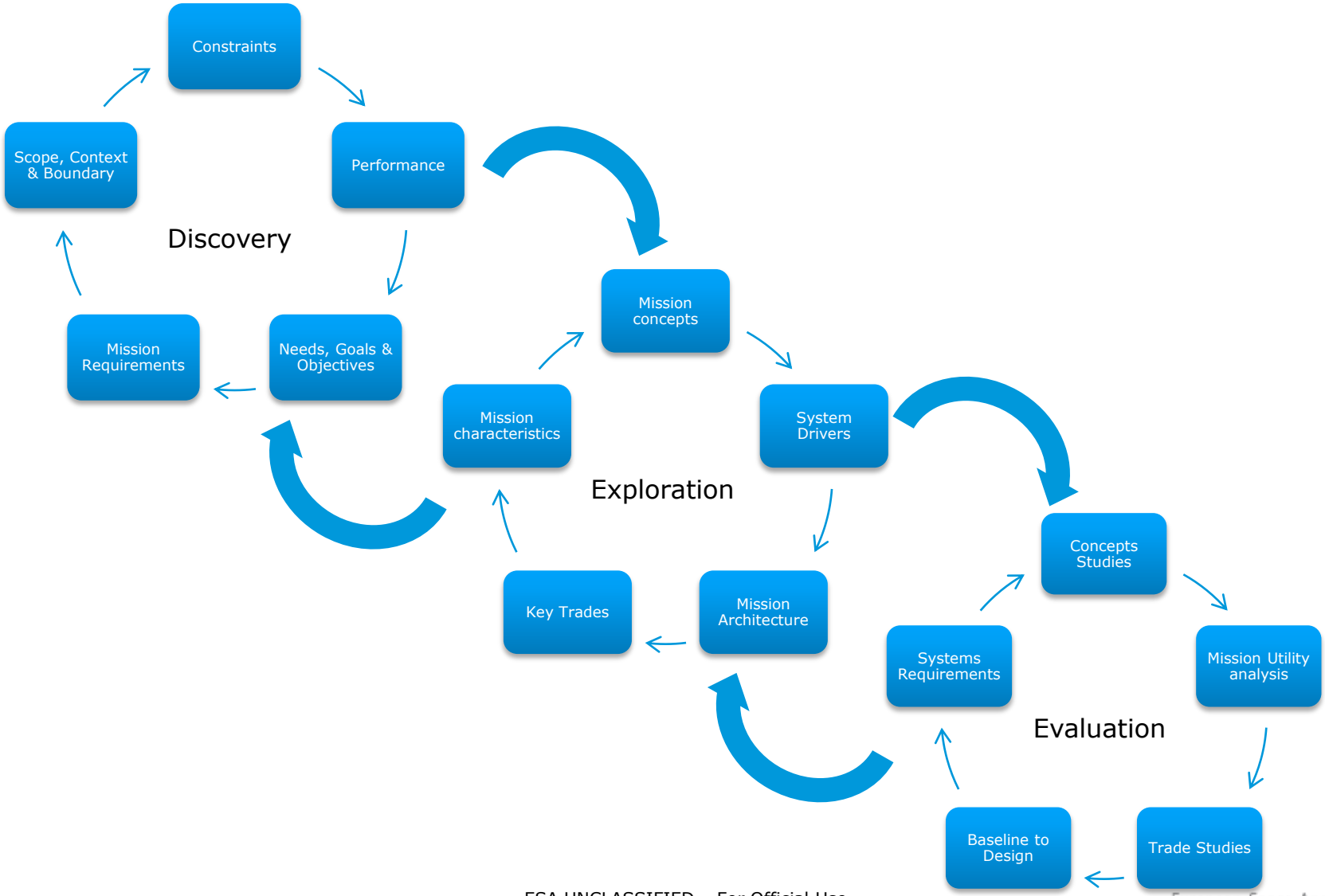
- Technical Requirements Specifications
- Interface Requirements Document
- Systems Engineering Plan
- Technical Budget
- Verification Plan
- Etc.
  See ECSS-E-ST-10C Annex A

Engineering a mission

- Need a system-level process to govern concept exploration, definition and development
  - In practice this is a non-linear process
  - It is iterated:
    - at each mission phase
    - at system and sub-system levels in each phase of implementation

- Must anticipate events and constraints during integration (e.g. manufacturability, assembly sequences) and verification
- Plan for deployment, use and end-of-life activities

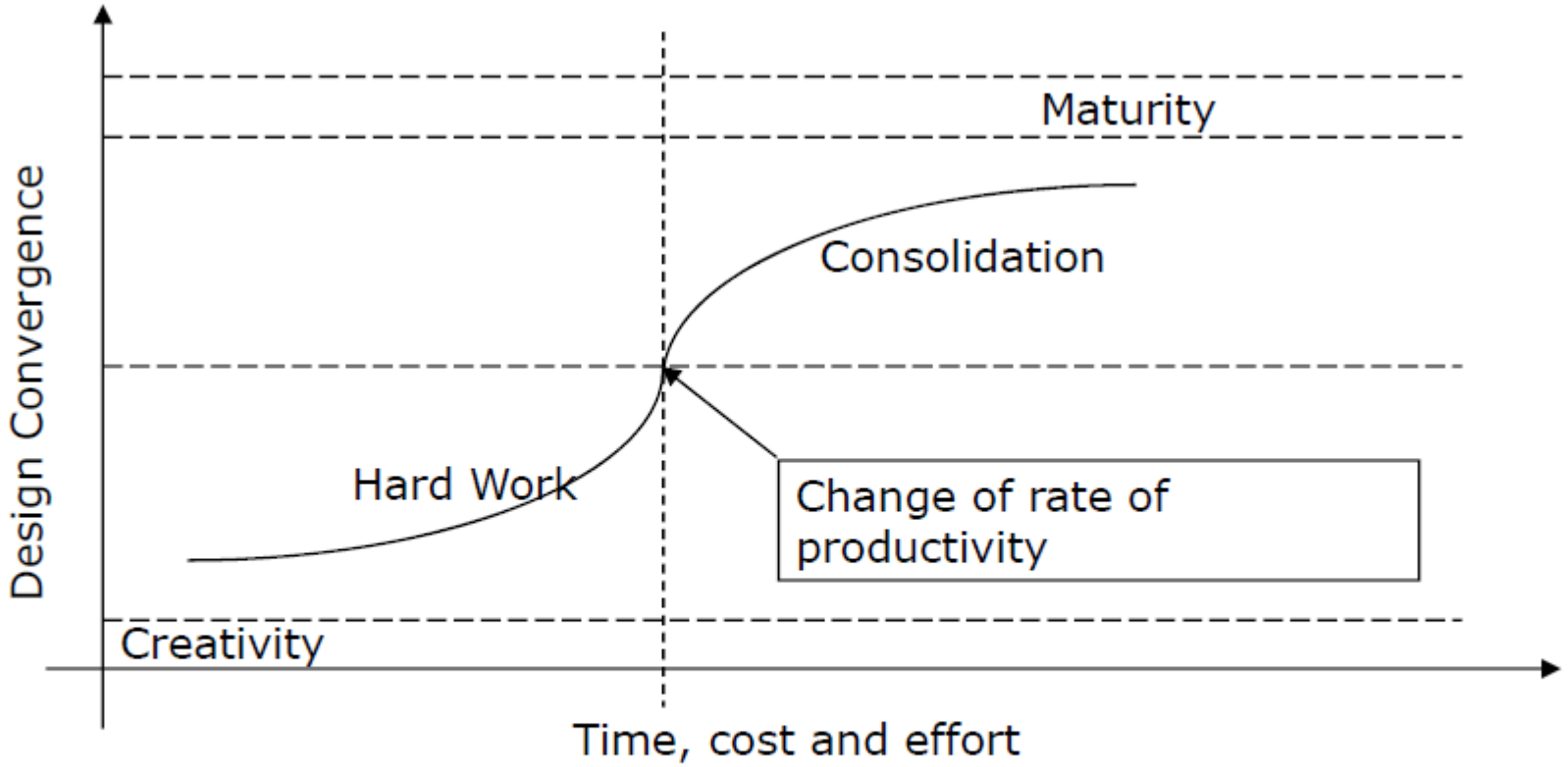
Iterative design cycles



Significant mission drivers

- Launcher needed (cost)
- Radiation environment
  - Shielding (cost, mass)
  - Electronic components (cost, availability)
  - Double/triple redundancy (cost, mass, complexity)
- Thermal environment
  - Heaters, insulation, radiators (cost, mass)
- Access to ground stations or data relay satellites
  - Comms system design (cost, mass, complexity)
- Insolation
  - Solar array size (cost, mass, complexity)
- Eclipse duration
  - Battery size (cost, mass)
- Launch date (cost, mass)

The S- curve



After a given point the iterations achieve only marginal improvement to the design
The system engineer shall understand when it is time to call the design cycle off

Iterations

- Iteration criteria for the design are:
  - Compliance to mission/system requirements
  - Compliance to available mass at launch
  - Compliance to cost target
  - Acceptability of mission technical and programmatic risks
- The tools to control iterations are:
  - Requirements compliance matrix
  - Spacecraft Mass budget
  - Cost budget
  - Risk tables
  - Other mission-specific performance indicators
- At the end of each design iteration the tools above are updated and evaluated. When they are all judged satisfactory the design is frozen

Design Process

- A good starting point in the design reduces iterations and design time
- It is the responsibility of the system engineer to provide a first set of starting points for the design (in addition to the mission/system requirements) to the team
- Whenever possible it is good practice to refer to previous missions with similar characteristics or initial requirements. A database of previous missions (and mistakes) is a very useful tool
- In other cases the system engineer will have to perform analyses on his/her own before initiating the design cycle

Design Process - Mass

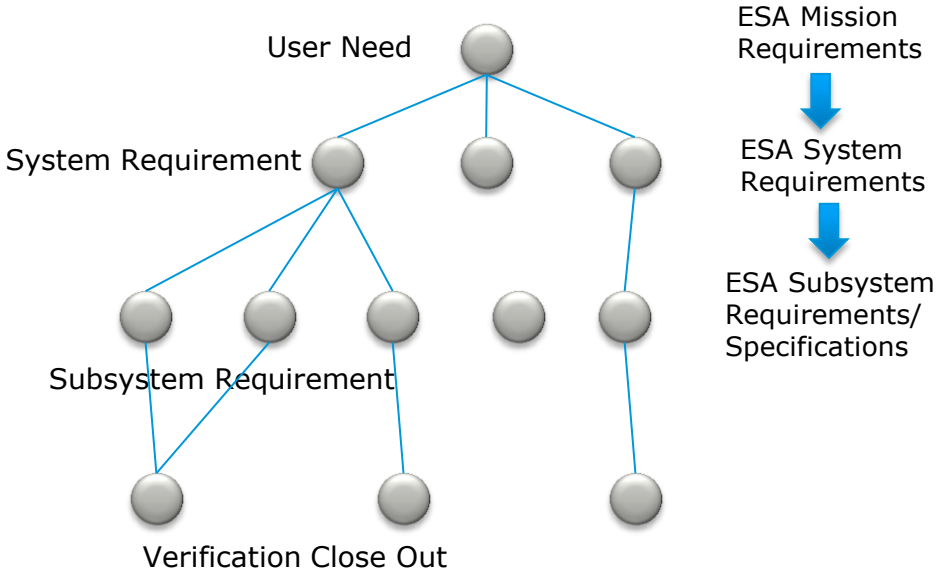
- The mass paradox:
  - The spacecraft dry mass is calculated bottom-up: from all subsystem units upwards and therefore requires a design of each subsystem
  - In order to design many subsystems the mass of spacecraft needs to be known (e.g propulsion, AOCS, thermal)
- How to start ?
  - For the first iteration the mass of the spacecraft is assumed to be = maximum launchable mass. The maximum available dry mass is then calculated from the delta-V budget.
  - The maximum available dry mass is then split among subsystems according to experience. This allows each subsystem to have a reference mass budget to work with for the first design iteration

Requirements Hierarchy

Requirement types	Respon.	Content	Document
User Requirements/ Mission Objectives	User	Non-technical, high level, general	URD or Mission Objectives Document
Mission Requirements	Customer/ Sponsor	Functional, technical, overall performance. Applies to the Mission	MRD or MSRD
System Requirements	Customer/ Sponsor	Functional, technical, overall performance. Applies to the System	SRD
System Requirements Specification	System Developer	Detailed, technical, reflects the design. Represents the interpretation of the customer req.s from the developer	System Specification
Lower level specifications	Lower Tier Supplier	Very specific and detailed, one-to-one correspondence to system requirements	Element, Subsystem or unit specification
Interface Requirements	Customer	Allows connecting the system with other systems	IRD and ICD
Operations Requirements	Operator	Constraints due to operations	OIRD

Requirements Process

- Aim is to translate user need
  - Language of the user
- Into a solution that can be designed and made
  - Language of the designer and manufacturer (coder etc.)



Requirements Engineering

Requirement engineering is defined on ECSS-E-ST-10-C and ECSS-E-ST-10-06C and includes:

- Requirement statement
- Requirement allocation
- Requirement validation
- Requirement maintenance

Requirements

- Types of requirements:
  - functional requirements,
  - mission requirements,
  - interface requirements,
  - environmental requirements,
  - operational requirements,
  - human factor requirements,
  - (integrated) logistics support requirements,
  - physical requirements,
  - product assurance (PA) induced requirements,
  - configuration requirements,
  - design requirements,
  - verification requirements.

Requirements Engineering

Formal statements of needs

3 key points to remember about **requirements**:

- define "what" is to be done
  ⇒ **function**
- define "how well" it has to be done
  ⇒ **performance**
- define (sometimes implicitly) how the requirement should be confirmed
  ⇒ **verification**

Requirements Constraints

Restrictions imposed to the possible design choices ("negative requirements"): e.g. operational, environmental, safety or regulatory constraints

Also of programmatic/politic/cost nature.

The most important are:

- Latest acceptable launch date
- Lowest level of technology development which is acceptable to the project (linked to cost and risk considerations) - TRL
- Launchers to be excluded (politics)
- Re-use or not of existing platforms or units (industrial policy/competition)

Requirements Allocation

- Requirements shall follow a hierarchy
- Requirements shall be linked to the mission/ product tree / Function tree
- Requirements need to be traceable: it must be possible to track a requirement for a subassembly up to higher levels by a logical/numerical link
- Requirements shall always be comparable to design/mission parameters
- Interface requirements shall be identified
- Requirements shall be maintained (docs and tools) *(i.e. Please don't use MS Word for req. engineering)*

Requirement Formulation

- Good requirements are:
  - Short/Synthetic
  - Definite/Unambiguous
  - Verifiable
  - Traceable
  - Formulated using terms that have been properly defined earlier
- Examples:

The system **shall** provide a *throughput* of *250 kbits/s*

The system total wet *mass* at launch **shall** *be less than 1300 kg*

Requirement Formulation

A TBC figure in a requirement means that the Author is proposing that figure but is asking the Contractor to perform its own analysis to confirm it: Violating a TBC req. is not a non-compliance

A TBD figure indicates that the Author has identified this as a relevant requirement but precise quantification is missing. It is left to the Contractor to propose and justify its own value for agreement with the Author.

A "should" requirement is a goal; meaning that first the "shall" req.s must be used to design the system and then, the impact of the additional "should" req. analysed. In case this impact is affordable, the "should" may be later turned into a "shall" (ECSS says "should" is a recommendation)

Requirement Formulation

Terms ***not*** to be used:

"As necessary"

"Appropriate"

"As far as possible"

"Optimise/minimise/maximise"

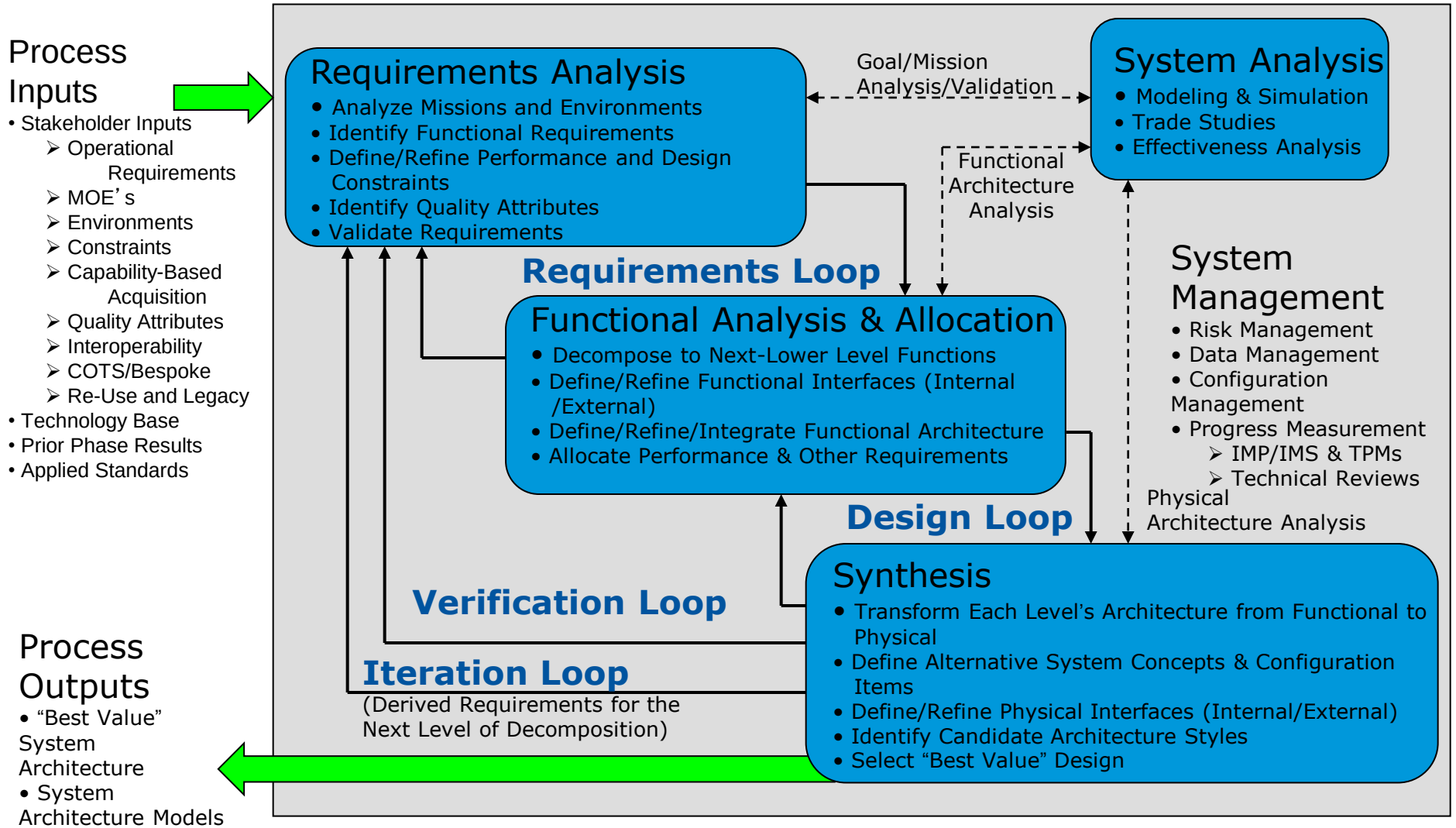
"Sufficient"

"Best possible"

"State-of-the-art"

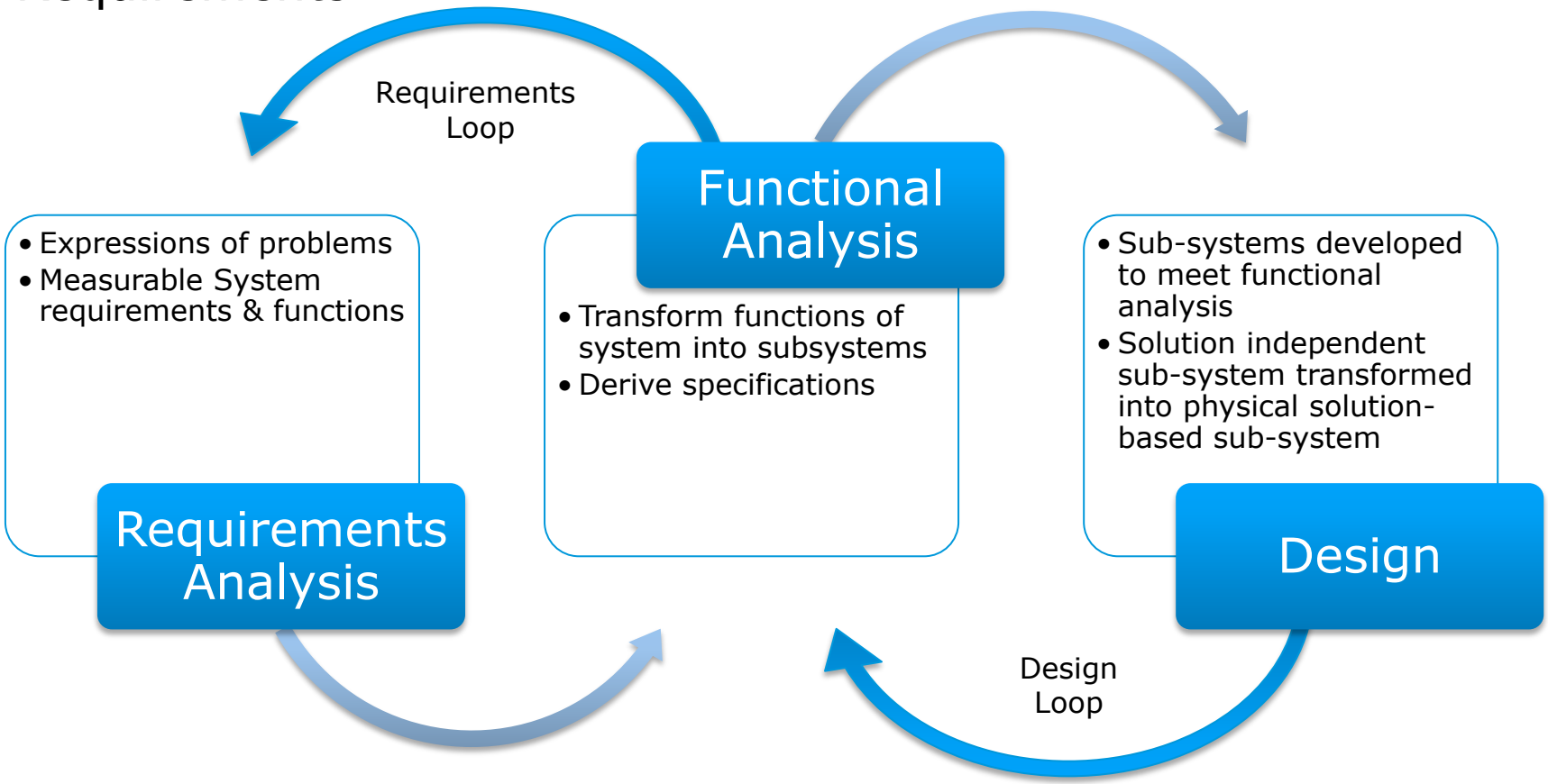
Etc…

Requirements to Design Process

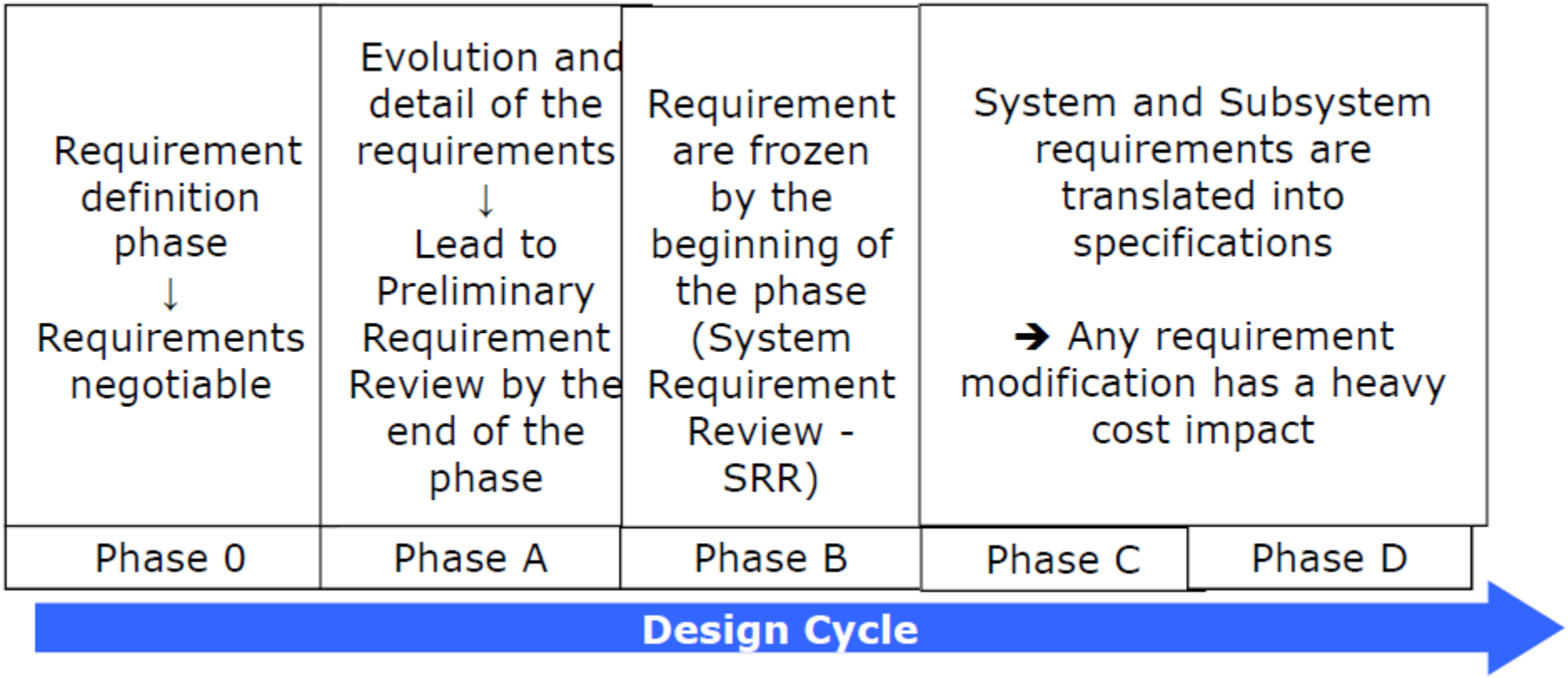


Requirements Loop

- Define, Derive, and Refine Functional/Performance Requirements

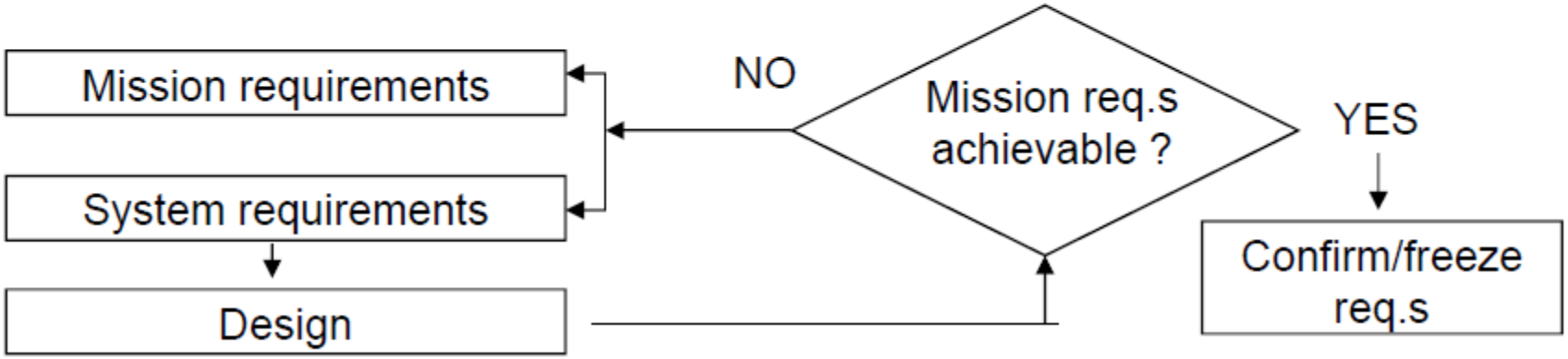


Requirements Evolution

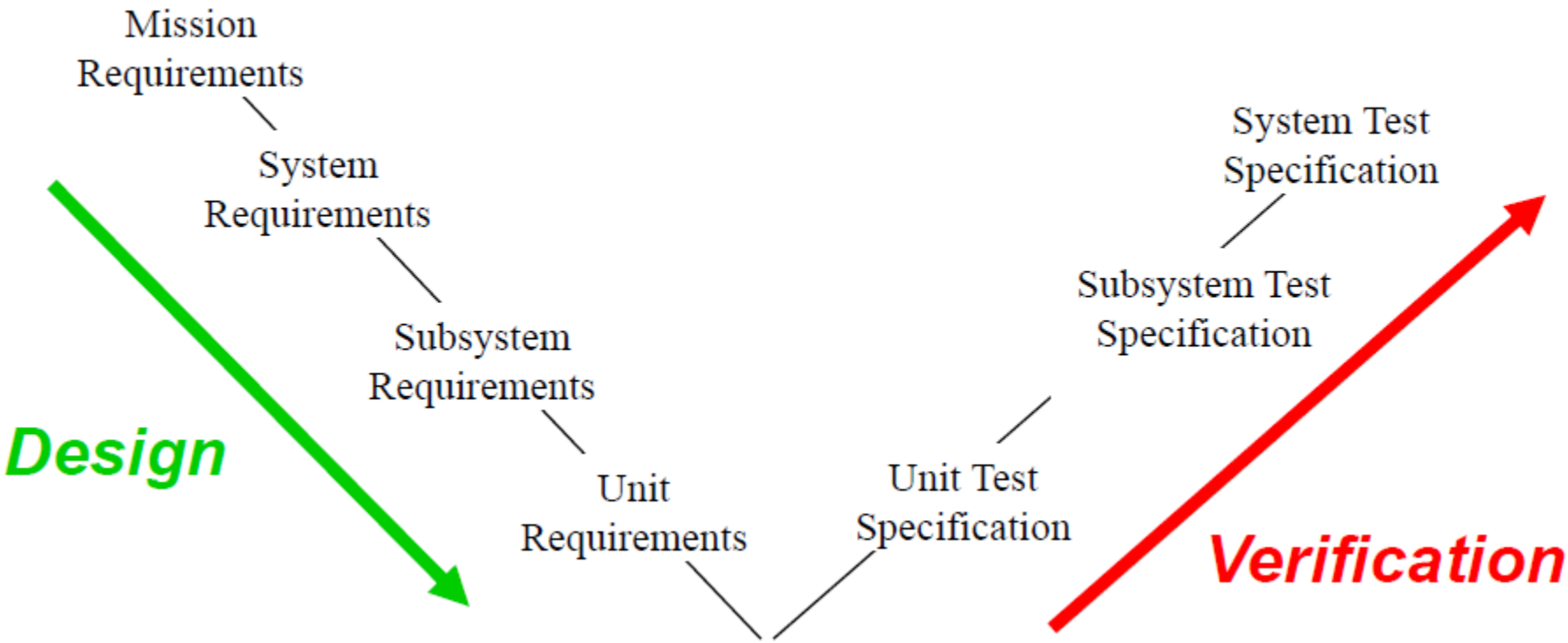


Requirements Iteration

Link between req.s and design in initial mission phases: req.s are allocated top-down but bottom-up design verification is used to confirm them



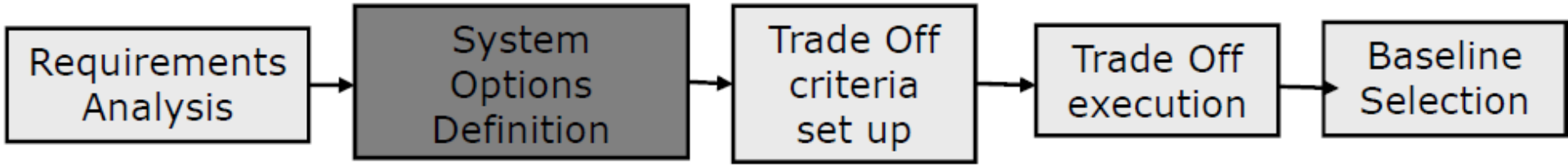
The V model


Mission Requirements
System Requirements
Subsystem Requirements
Unit Requirements
Unit Test Specification
Subsystem Test Specification
System Test Specification
Design
Verification

System Interfaces

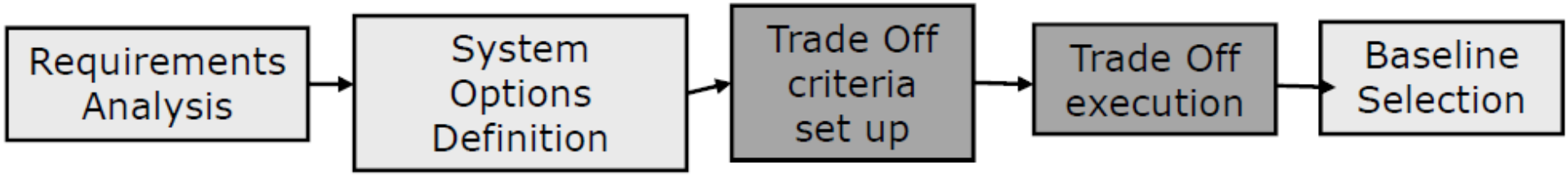
- Interfaces result from how a system is defined and arranged (boundaries)
- Have at least 2 interfaces:
  - between each pair of components or subsystems
  - between the system and the external environment
  - But real systems have many internal and external interfaces that need to be understood for the system to perform as desired
- Interfaces can be:
  - static, such as the walls of a building
  - Or *dynamic,* such as a data bus.
- Interfaces may change with time:
  - Designed to be flexible
  - System is able to evolve – its parts can reconfigure themselves
    - Such as swarms of bees, soldiers, footballers
  - The architect redefines the system
  - A new architect (re)defines the system
- Information usually captured in an Interface Control Document (ICD)
- Configuration control and formal change management is required to manage the information contained within the ICDs

Mission and System Options



- Mission/System options are direct consequences of the Requirement Analysis (different perceived ways to comply with the requirements)
- Exhaustive search of all possible system options has to be done in the early stages of a project
  → In order to schematize, the options shall be organized on trade tree or a matrix (systematic)
- Option selection is performed via trade offs (supported by the previous trade tree)

Mission and System Trade-Offs



- Trade off shall be established for option selection in a clear way:
  - Trade off criteria
  - Relative weighting
- Trades shall be based as much as possible on quantitative criteria (but in early phases qualitative/parametric trades are common)
- Trade offs shall be well documented and recorded, as they may be re-opened in later phases
- Option selection will end with the definition of baseline and back up option(s)

Trade-Off Selection

Parameters	Weighting factors	Option 1		Option 2		Option 3	
		Ranking [1 - 5]	Score	Ranking [1 - 5]	Score	Ranking [1 - 5]	Score
Total Mass	0.3	4	1.2	5	1.5	2	0.6
Payload Mass	0.3	2	0.6	4	1.2	2	0.6
Cost	0.2	3	0.6	1	0.2	5	1
Risk	0.1	2	0.2	2	0.2	4	0.4
Complexity	0.1	3	0.3	4	0.4	4	0.4
	1		**2.9**		**3.5**		**3**

Notes:

- Weighting factor: from 0 (less important) to 1 (more important)
  (Overall sum should be equal to 1)
- Ranking: From 5 (better) to 1 (worst). Scale is arbitrary.

Main System Budgets

Overall System synthesis is based on budgets:

- Mass budget
- Delta-V and Propellant budget
- Power/Energy budget
- Data budget
- Link budget
- Pointing error budget
- Cost

Budget Margins

- ΔV margin, at least 5% (higher if gravity losses are not accounted for)
- Data processing margin
  - 50% for mass memory
  - 100% for computing power
- Communications, 3 dB in the link budget
- Temperatures, ±10 deg
- Power, 20% in the power budget
- Consumables (AOCS propellant, battery, to be sized for mission lifetime + extension)
- Propellant residuals (unusable): 3% of propellant mass

Budget Margins

- Always have robust margins but be careful to oversize your system.
- Lot of pressure to reduce margin to have more payload
- Margins evolve with mission phases a typical evolution is:
  - 20% at SRR
  - 15 % at PDR
  - 10% at CDR
  - 5% or less at FAR

Main Documentation and Design Outputs

- Requirements
  - Mission/User Reqs => System Reqs => Segment Reqs => Facility/Subsystem Reqs => Component Reqs
- Design
  - Design Definition File (DDF)
    - Functional Architecture
    - Logical Architecture
    - System Budgets
    - Interface Definitions
  - Design Justification File (DJF)
    - Trade-offs