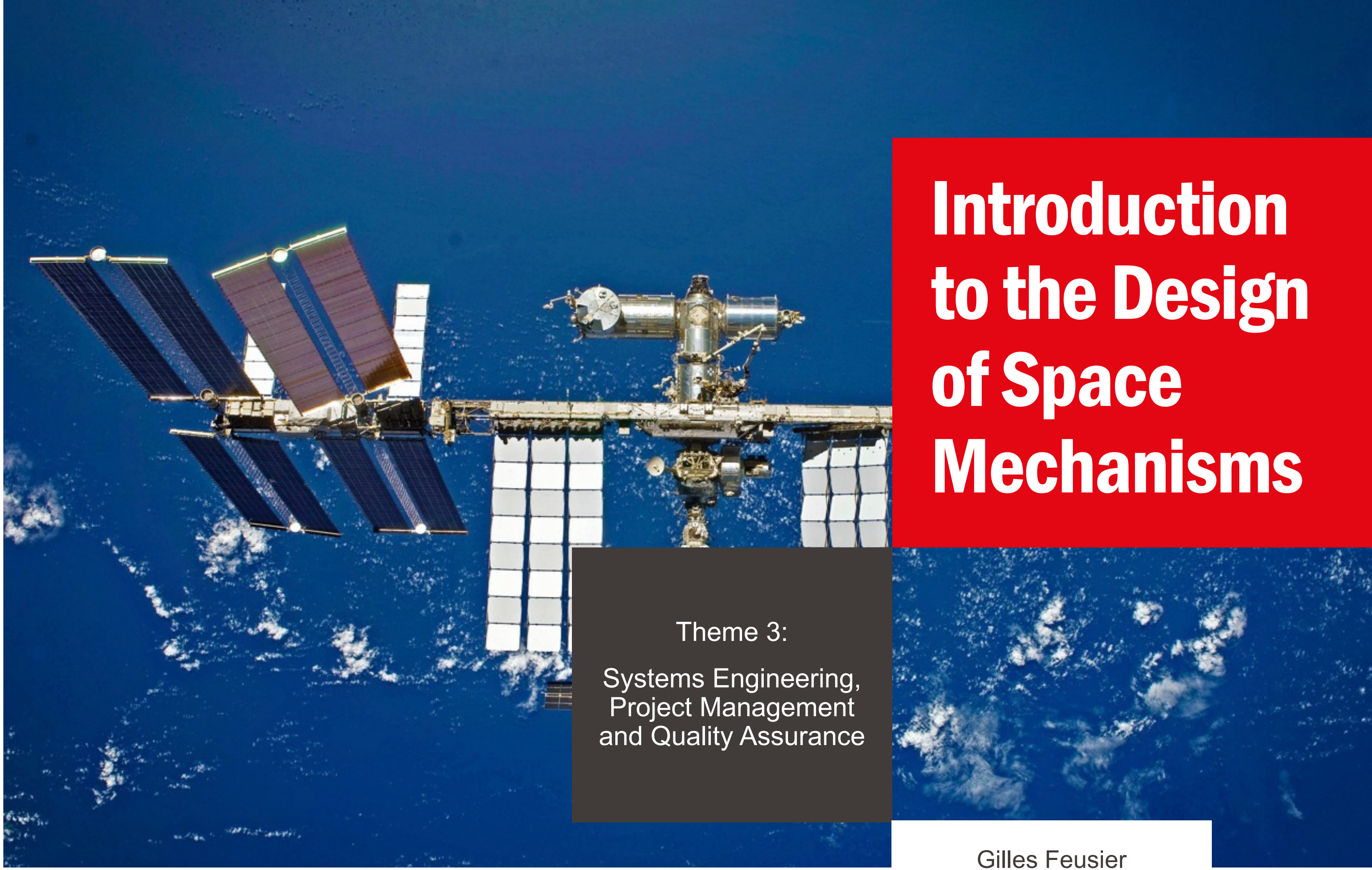
# Introduction to the Design of Space Mechanisms



Theme 3:

Systems Engineering, Project Management and Quality Assurance

Gilles Feusier

# Space Mechanism Projects



“Fundamentals of Systems Engineering”
By Prof. Olivier L. de Weck MIT/EPFL ENG-421

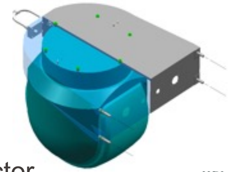
- **Systems Engineering**

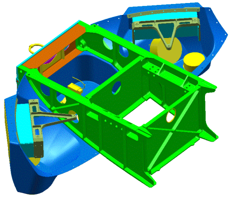
  NASA-SP-2016-6105 Rev 2 Systems Engineering Handbook:
  - At NASA, “systems engineering” is defined as a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system.
  - A “system” is the combination of elements that function together to produce the capability required to meet a need.

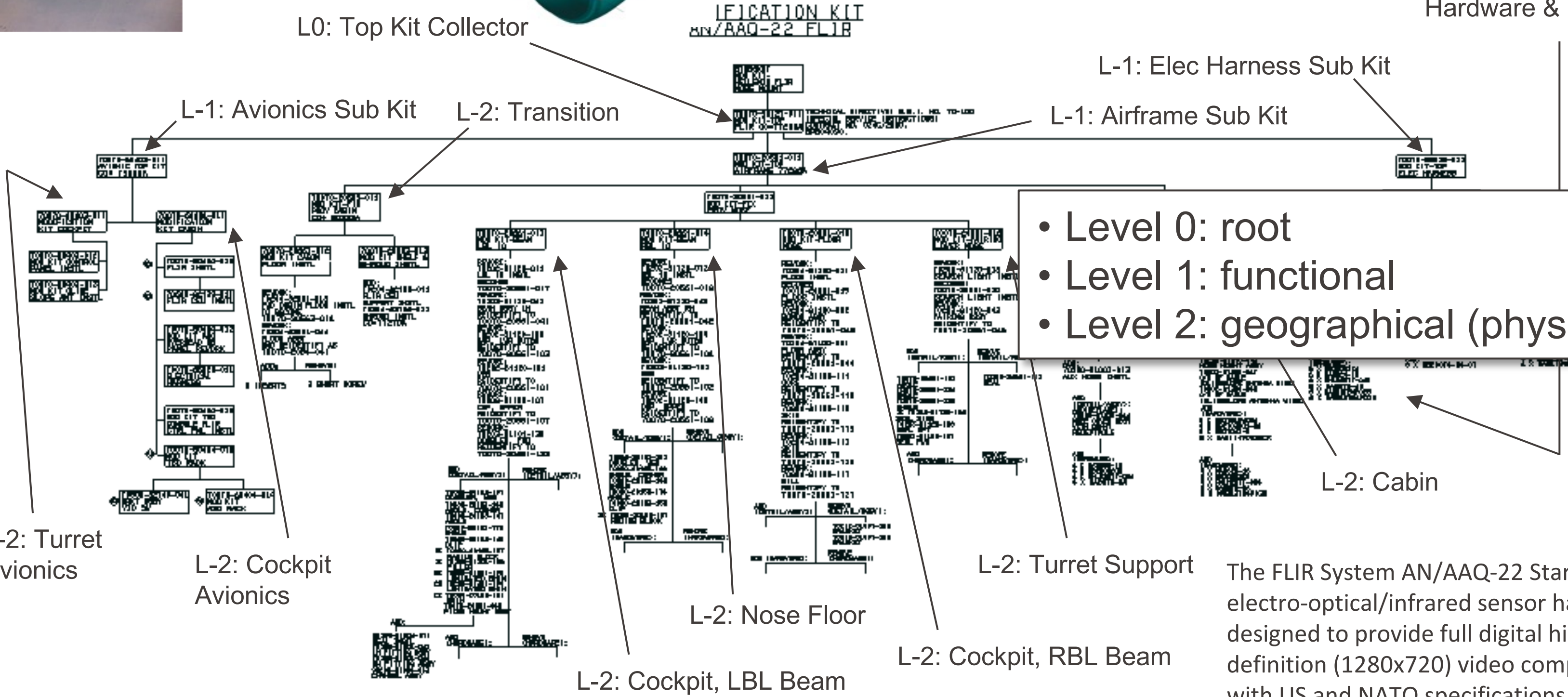
  ECSS-E-ST-10C Rev.1 System engineering general requirements:
  - A “system” is a set of interrelated or interacting functions constituted to achieve a specified objective (ECSS-S-ST-00-01C).
  - A “subsystem” is a part of a system fulfilling one or more of its functions (ECSS-S-ST-00-01C).
- **Project Planning**
- **Project Documentation**
- **Quality Assurance (QA), Product Assurance (PA)**

# Example: FLIR System for Aircraft

FLIR = Forward Looking Infrared








- Level 0: root
- Level 1: functional
- Level 2: geographical (physical)

The FLIR System AN/AAQ-22 Star SAFIRE electro-optical/infrared sensor has been designed to provide full digital high-definition (1280x720) video compliant with US and NATO specifications.

EPFL

# Why do we need system decomposition?



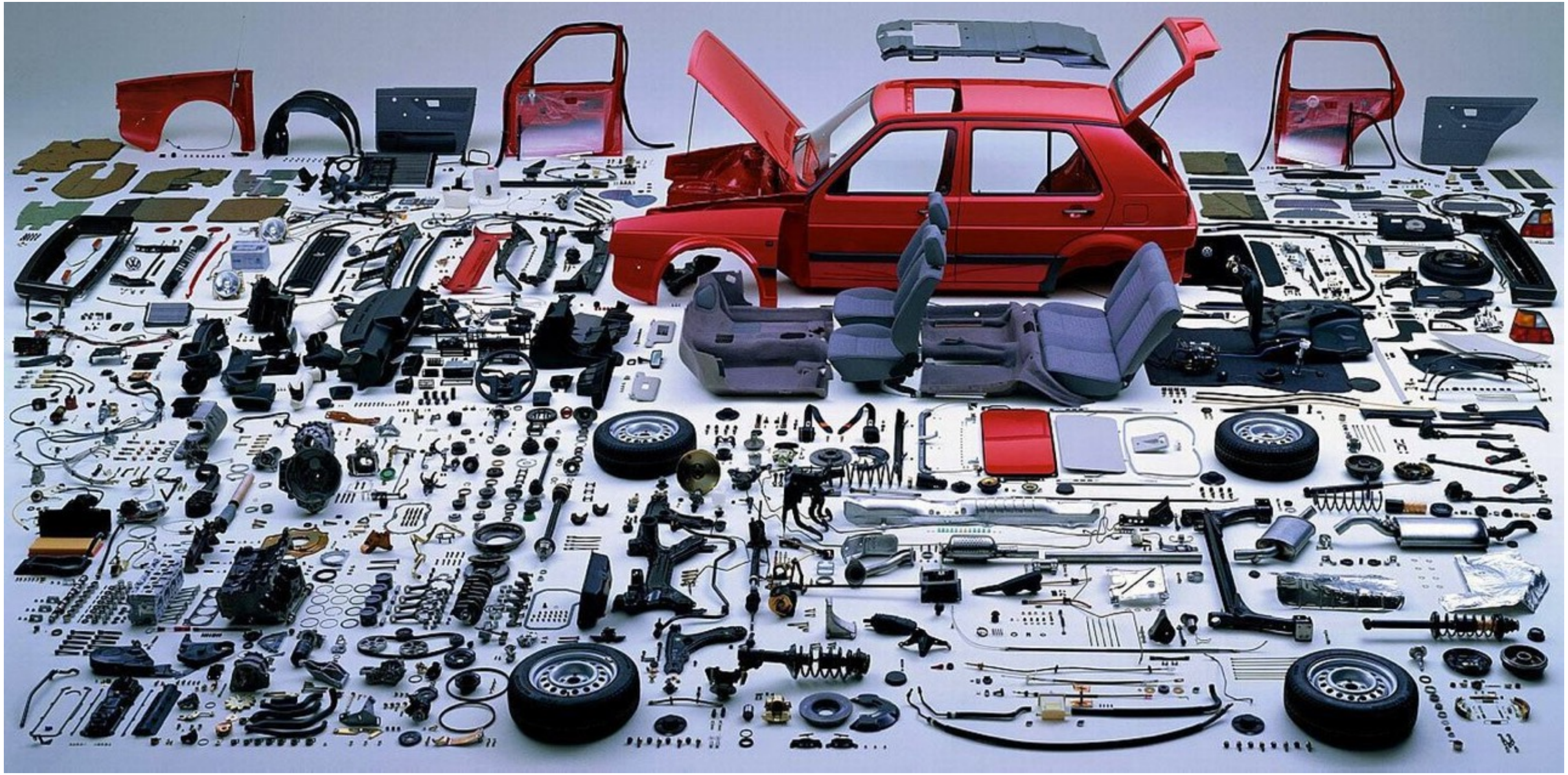


Image Source: http://www.teslamotorsclub.com/showthread.php/29692-How-many-moving-parts-in-a-Model-S/page3

# System Complexity

- How many levels  in drawing tree?

Assume 7-tree [Miller 1956]
http://www.musanim.com/miller1956/

$$\# \; levels = \left\lceil \frac{\log(\# \; parts)}{\log(7)} \right\rceil$$

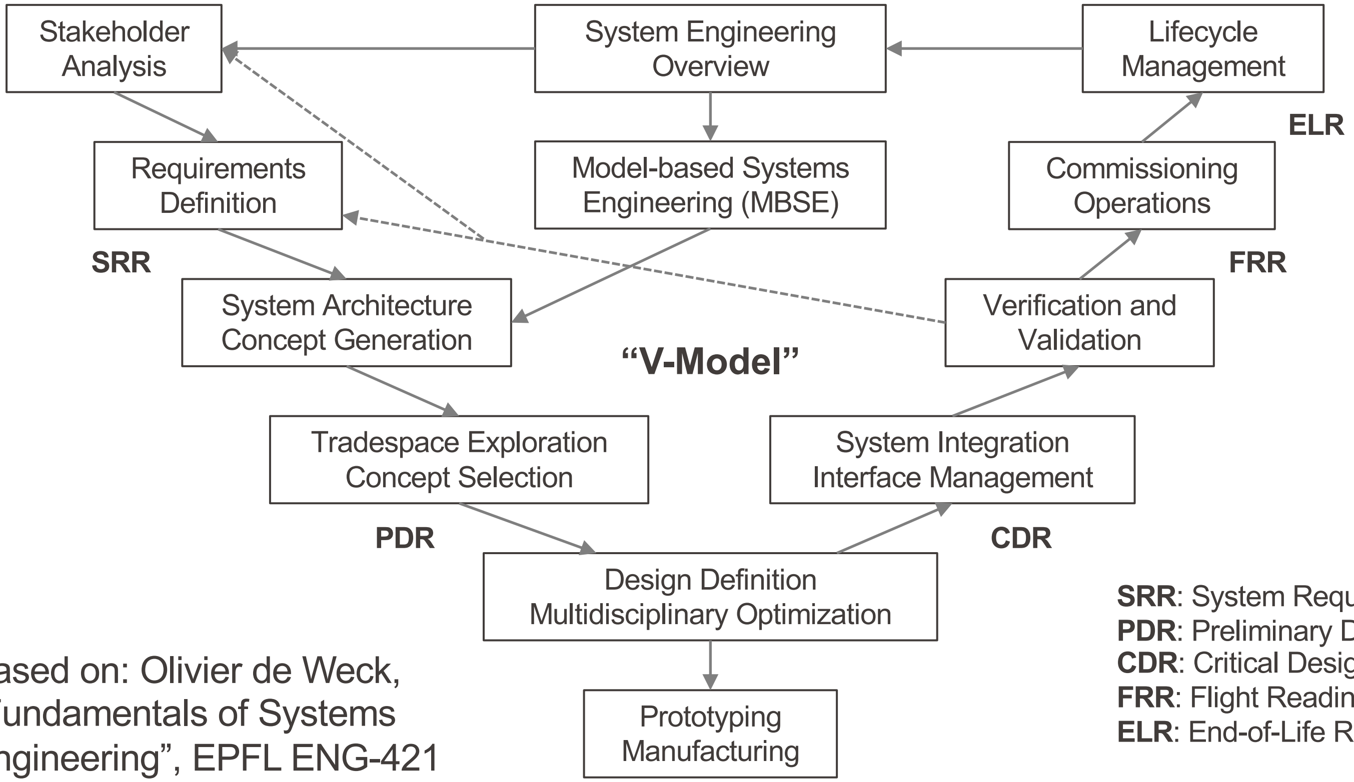
| | | ~ # parts | # levels |
|---|---|---|---|
| • Screwdriver | (B&D) | 3 | 1 |
| • Roller Blades | (Bauer) | 30 | 2 |
| • Inkjet Printer | (HP) | 300 | 3 |
| • Copy Machine | (Xerox) | 2’000 | 4 |
| • Automobile | (GM) | 10’000 | 5 |
| • Airliner | (Boeing) | 100’000 | 6 |

simple

complex

*Source: Ulrich, K.T., Eppinger S.D. , Product Design and Development Second Edition, McGraw Hill, 2nd edition, 2000, Exhibit 1-3*

# The famous "V-Model" of Systems Engineering




Based on: Olivier de Weck, "Fundamentals of Systems Engineering", EPFL ENG-421

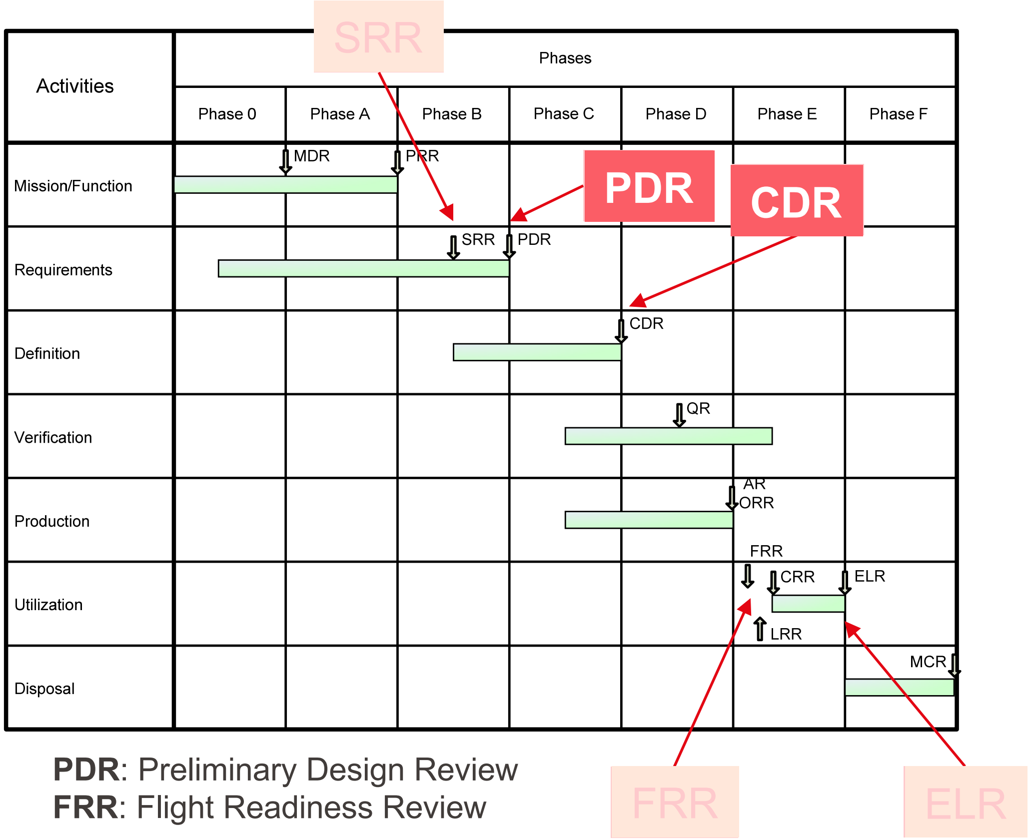
**SRR**: System Requirements Review
**PDR**: Preliminary Design Review
**CDR**: Critical Design Review
**FRR**: Flight Readiness Review
**ELR**: End-of-Life Review

Note: reviews acc. to ECSS-M-ST-10C Rev. 1

# The lifer cycle of space projects - ESA

**ECSS-M-ST-10C**
Project planning and implementation

- Phase 0 - Mission analysis/needs identification
- Phase A - Feasibility
- Phase B - Preliminary Definition
- Phase C - Detailed Definition
- Phase D - Qualification and Production
- Phase E - Utilization
- Phase F - Disposal




**SRR**: System Requirements Review
**CDR**: Critical Design Review
**ELR**: End-of-Life Review
**PDR**: Preliminary Design Review
**FRR**: Flight Readiness Review

# SE Standards and Handbooks

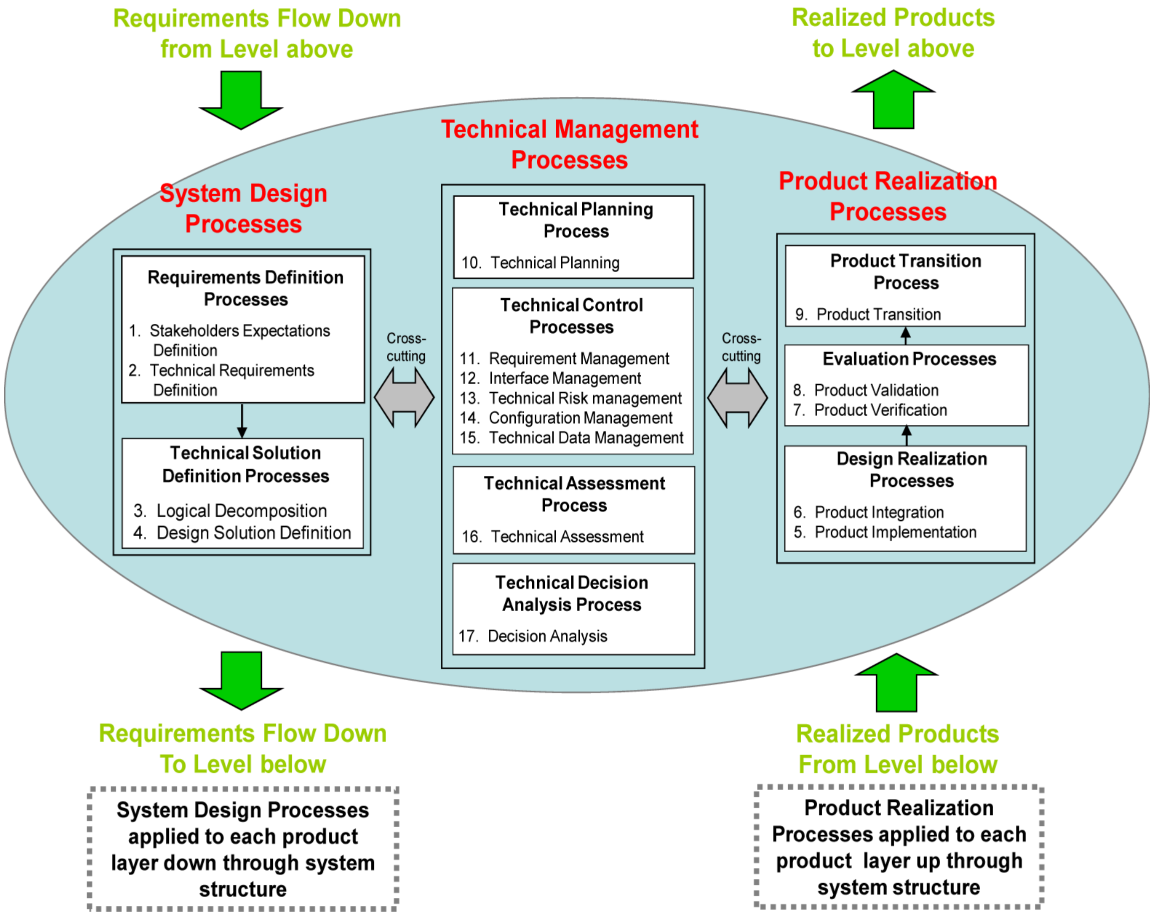
- **Systems Engineering Standards**
  - **ECSS-E-ST-10C Rev.1** – European Systems Engineering Standard, http://www.ecss.nl/
  - **NASA Systems Engineering Handbook**, NASA/SP-2016-6105, Rev 2, 2016
  - **INCOSE Systems Engineering Handbook**, A Guide for System Lifecycle Processes and Activities, version 4, International Council on Systems Engineering (INCOSE), June 2023
  - **ISO/IEC/IEEE 15288:2023**, Systems and software engineering - System life cycle processes; Ingénierie des systèmes et du logiciel - Processus du cycle de vie du système – May 2023 edition

# Questions to be asked



- **Why are we doing the project?** → **Stakeholder Analysis**
- **What must we achieve?** → **Requirements Definition**
- **How could we do it?** → **System Architecture & Concept Generation**
  - Oftentimes there are many different ways

# The NASA Systems Engineering "Engine"




From
**NASA Procedural Requirements**

*NPR 7123.1B*

# Requirement Definition

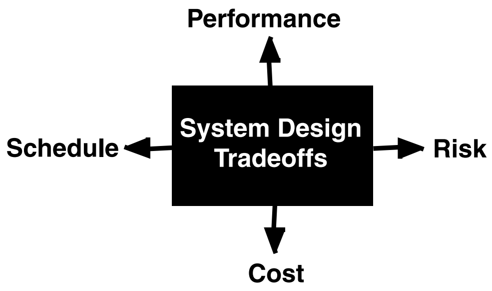
- Requirements describe the **necessary functions and features of the system** we are to conceive, design, implement and operate.
  - Performance
  - Schedule
  - Cost
  - Other Characteristics (e.g. lifecycle properties)
- Requirements are often organized hierarchically
  - At a high-level requirements focus on **what** should be achieved, not how to achieve it.
  - Requirements are specified at every level, from the overall system to each hardware and software component.
- **Critically important to establish properly**
  - Many of the cost overruns are caused by over-ambitious or missing requirements

# Requirements set constraints and goals

- When designing systems we always have tradeoffs between performance, cost, schedule and risk

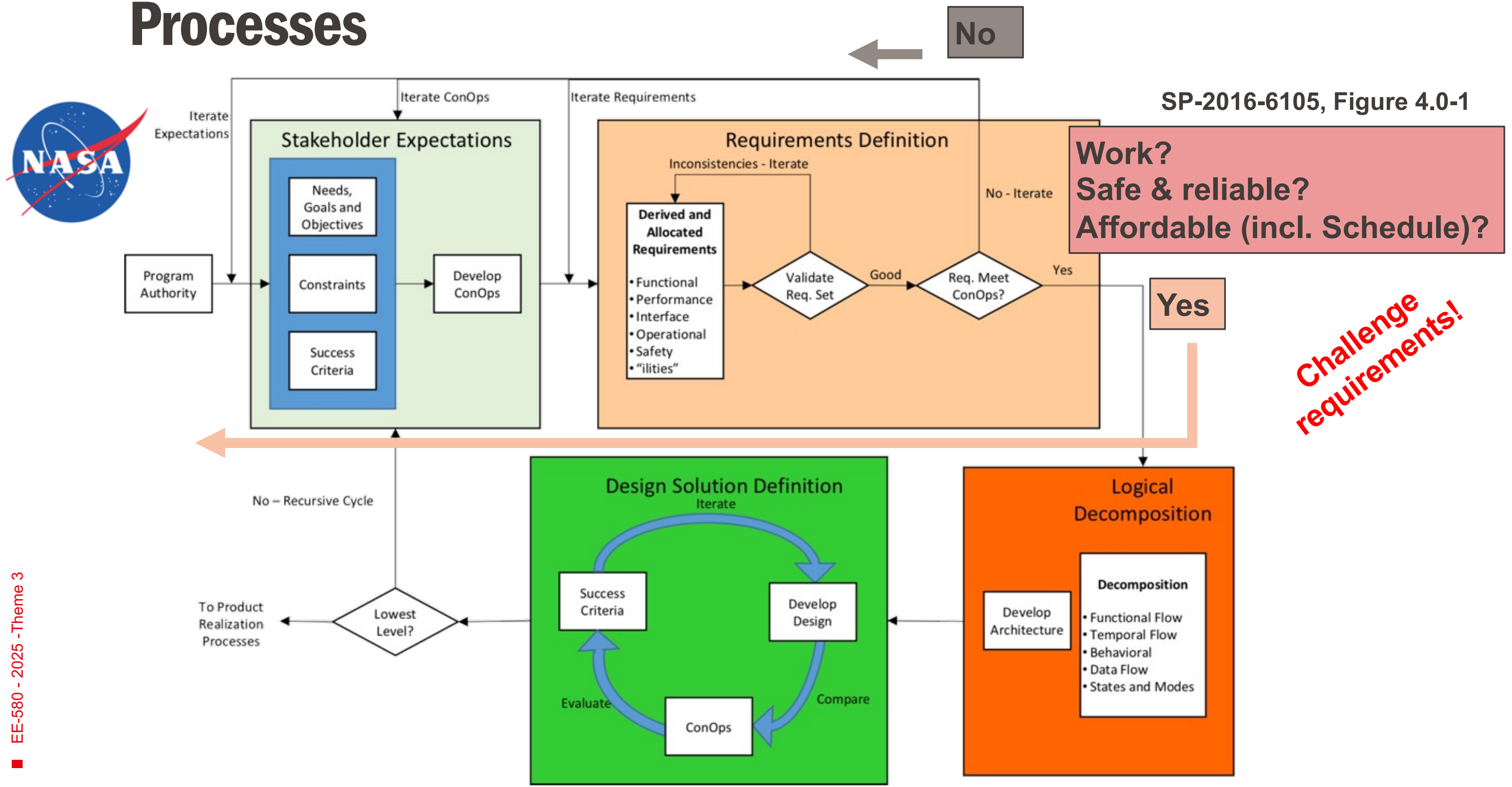
- "**Shall**" … Requirements help set constraints and define the boundaries of the design space and objective space

- "**Should**" … Requirements set goals once "shall" requirements are satisfied



# Relationships among the upstream System Design Processes

SP-2016-6105, Figure 4.0-1



# Attributes of Acceptable Requirements

- **A complete sentence with a single "shall" per numbered statement**
- **Characteristics for each Requirement Statement:**
  - Clear and consistent – readily understandable
  - Correct – does not contain error of fact
  - Feasible – can be satisfied within natural physical laws, state of the art technologies, and other project constraints
  - Flexibility – Not stated as to how it is to be satisfied
  - Without ambiguity – only one interpretation makes sense
  - Singular – One actor-verb-object requirement
  - Verify – can be proved at the level of the architecture applicable
- Characteristics for pairs and sets of Requirement Statements:
  - Absence of redundancy – each requirement specified only once
  - Consistency – terms used are consistent
  - Completeness – usable to form a set of "design-to" requirements
  - Absence of conflicts – not in conflict with other requirements or itself

# Common problems with requirements

- **Writing implementations (“How”) instead of requirements (“What”)**
  - Forces the design
  - Implies the requirement is covered

- **Using incorrect terms**
  - Avoid “support”, “but not limited to”, “etc”, “and/or”

- **Using incorrect sentence structure or bad grammar**

# Common problems (continued)

- **Writing unverifiable requirements**
  - E.g., minimize, maximize, rapid, user-friendly, easy, sufficient, adequate, quick

- **Missing requirements**
  - Requirement drivers include:

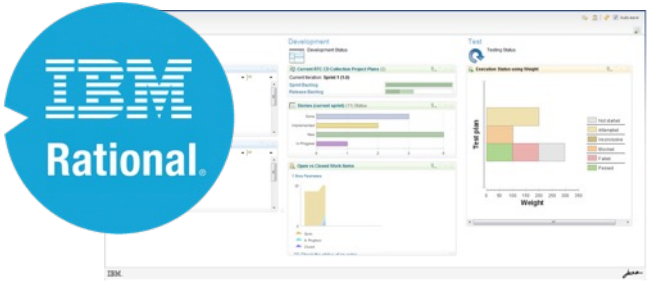
| | | |
|---|---|---|
| Functional | Performance | Interface |
| Environment | Facility | Transportation |
| Training | Personnel | Reliability |
| Maintainability | Operability | Safety |

- **Requirements only written for "first use"**

- **Over-specifying**

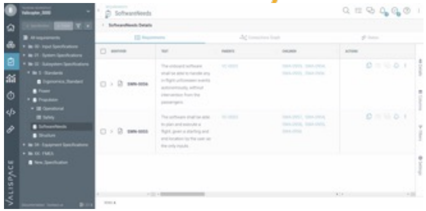
# Requirements Capture: Documents vs. Database

- Where / how are requirements captured?
  - Low cost “easy” solution: Create a document (e.g. in MS Word or Excel) to capture and revise the requirements. Use hyperlinks to link requirements.
  - This works okay for smaller projects with dozens or a few hundred requirements (e.g. about 3 levels of decomposition ➔ $(7+/-2)^3$ = 125 to 729
  - For larger projects with >1’000 requirements need to use a relational database
  - Commercial Tools, e.g. DOORS, Valispace, ReqView … are available (but can be expensive)




IBM Rational DOORS
http://www.ibm.com/developerworks/rational/library/rational-doors-next-generation-getting-started/tutorial/index.html

VALISPACE



Valispace
https://www.valispace.com


ReqView
https://www.reqview.com

# Example of Requirement (document type)

swiss space center

Doc. ref.: 0006_ES_007_SSC
Issue : 2 Rev : 1
Page : 1 of 11

**_SpaceCam_**
Requirements

*Prepared by:*
**Name**
G. Feusier

*Checked/App*
**Name**
G. Feusier

swiss space center

Doc. ref.: 0006_ES_007_SSC
Issue : 2 Rev : 1
Page : 4 of 11

## 6 Interfaces Requirements

### 6.1 Mechanical Interfaces

[6.1.1] The overall dimensions of the SpaceCam shall not exceed the values provided in Figure 1 [AD1].

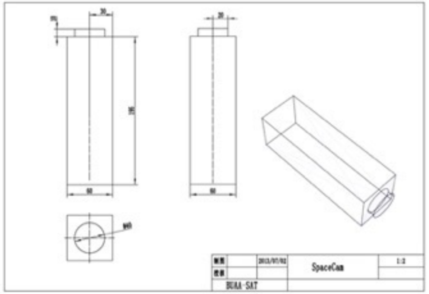



**Figure 1 – Dimensions of the SpaceCam**

[6.1.2] The mechanical interfaces shall be according to [RD1].

[6.1.3] The mass of the SpaceCam shall be lower or equal to 1.4 kg.

[6.1.4] The BUAA-SAT plate naming convention described in Figure 2 shall be used in the

swiss space center

Doc. ref.: 0006_ES_007_SSC
Issue : 2 Rev : 1
Page : 5 of 11

### 6.2 Thermal Interfaces

[6.2.1] The SpaceCam shall operate at the temperatures given in Figure 3 and Figure 4.

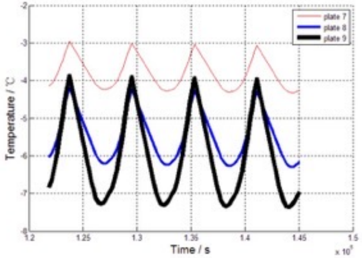



**Figure 3 – Cold case ten**

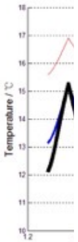



**Figure 4 – Hot case tem**

### 6.3 Electrical Interface

swiss space center

Doc. ref.: 0006_ES_007_SSC
Issue : 2 Rev : 1
Page : 9 of 11

**Table 6 – Sine vibration test level for acceptance (LM-2C/CTS [AD1]).**

| | Frequency [Hz] | Test load |
|---|---|---|
| Longitudinal | 5-10 | 2.5mm |
| | 10-100 | 1.0g |
| Lateral | 5-10 | 1.75mm |
| | 10-100 | 0.7g |
| Scan rate | | 4 Oct./min |

[8.1.3] Sine vibration frequency tolerance shall be ≤ ±2%.

[8.1.4] Sine vibration amplitude tolerance shall be ≤ ±10%.

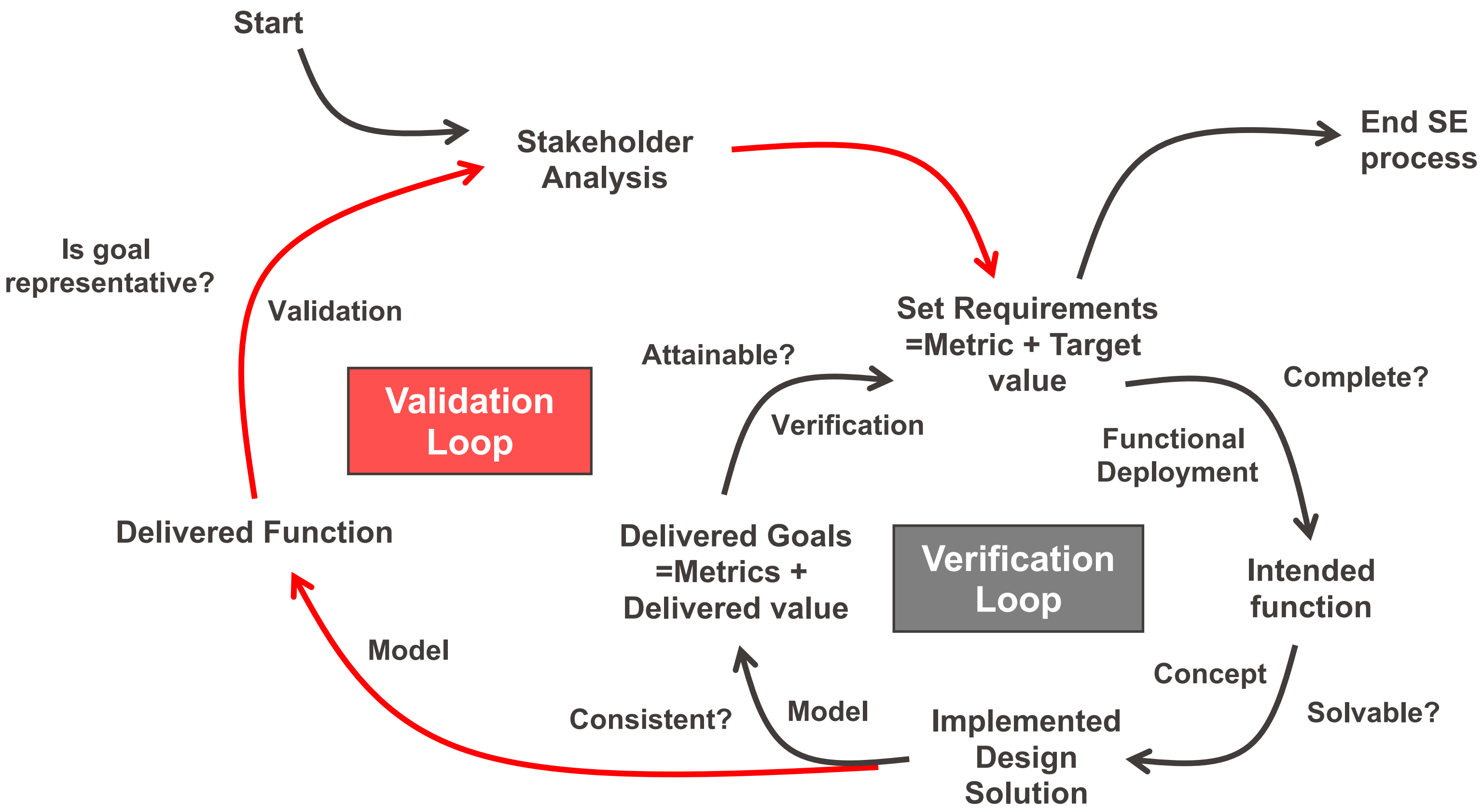
# Verification



- **Every requirement must be verified to ensure that the proposed design actually satisfies the requirement by** (ECSS-E-ST-10-02C):
  - Test (including demonstration)
  - Analysis (including similarity)
  - Review-of-Design (ROD),
  - Inspection

- **Requirements documentation specifies the development phase and method of verification**

# Verification and Validation Loops



# Types of Documents (non exhaustive)

- Inputs
  - Statement of Work (SoW), contract
  - Requirements
  - Standards and other reference or applicable documents
  - …
- Output
  - Proposal (technical, financial, management)
  - Matrix of Conformity
  - Configuration Management: configuration item data list (CIDL)
  - Technical Description, including justifications
  - Risk Analysis
  - Failure Modes, Effects and Criticality Analysis (FMECA)
  - Test plan and test procedures
  - Part List and Drawings
  - Declared Material and Process Lists
  - As-built status: as-built configuration data list (ABCL)
  - Non-conformances, requests for waiver
  - …

→

- *Digitalization (Space 4.0)*
- *Model Based Systems Engineering (MBSE)*
- *Electronic Data Sheets*
- *…*

**ECSS-M-ST-10C Rev. 1**
*Project planning and implementation*

**ECSS-M-ST-40C Rev. 1**
*Configuration and information management*

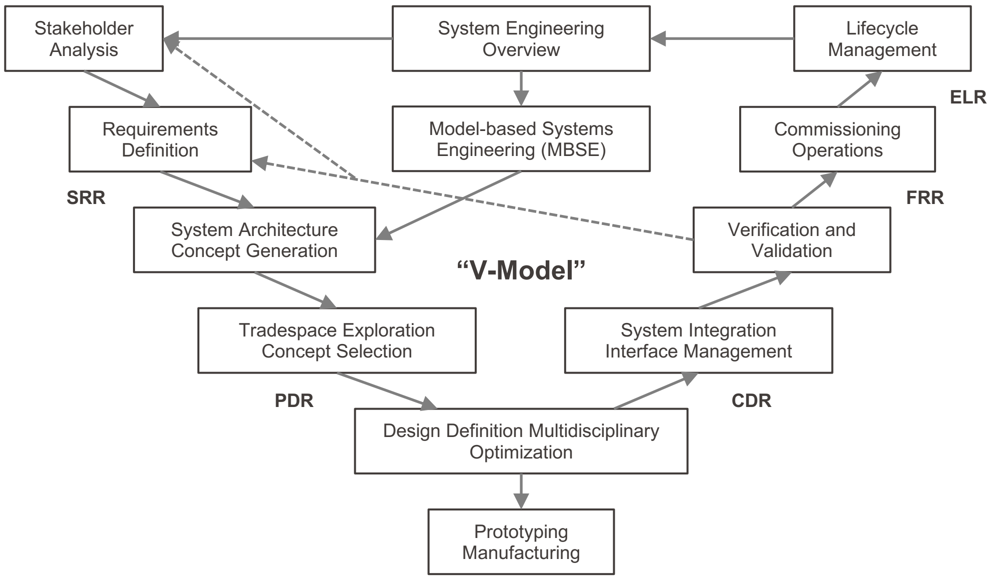
# Quality Assurance and Documentation

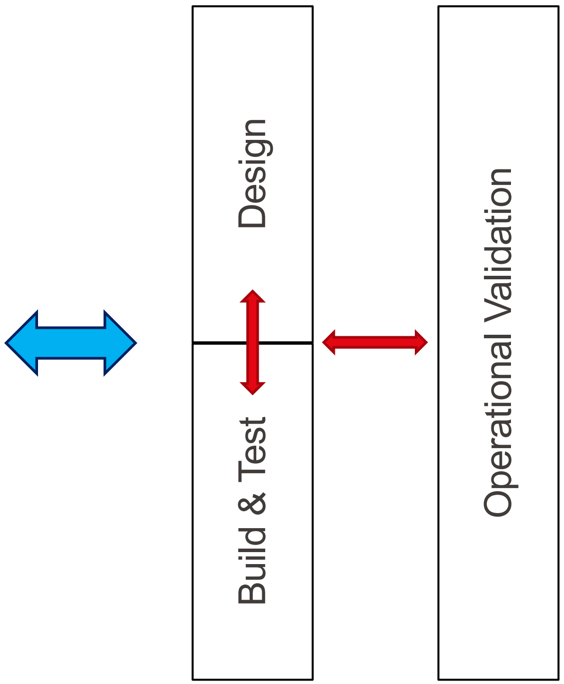
- Very detailed documentation and development processes
  - Defined in ECSS for ESA projects (ECSS-S-ST-00C)
  - Similar (but not identical) for NASA or others
- The documentation permits to control the risks
  - Technical risks of the project
    - Complexity
    - Existing technologies
    - Constraints and physical limitations
    - Other factors
  - Realization of the project
    - Products to be supplied
    - Required resources
    - Task to be completed
  - Schedule
  - Costs

**ECSS-Q-ST-10C Rev.1 - Product assurance management**

*"The prime objective of Product Assurance is to ensure that space products accomplish their defined mission objectives in a safe, available and reliable way."*

# Other/complementary approaches

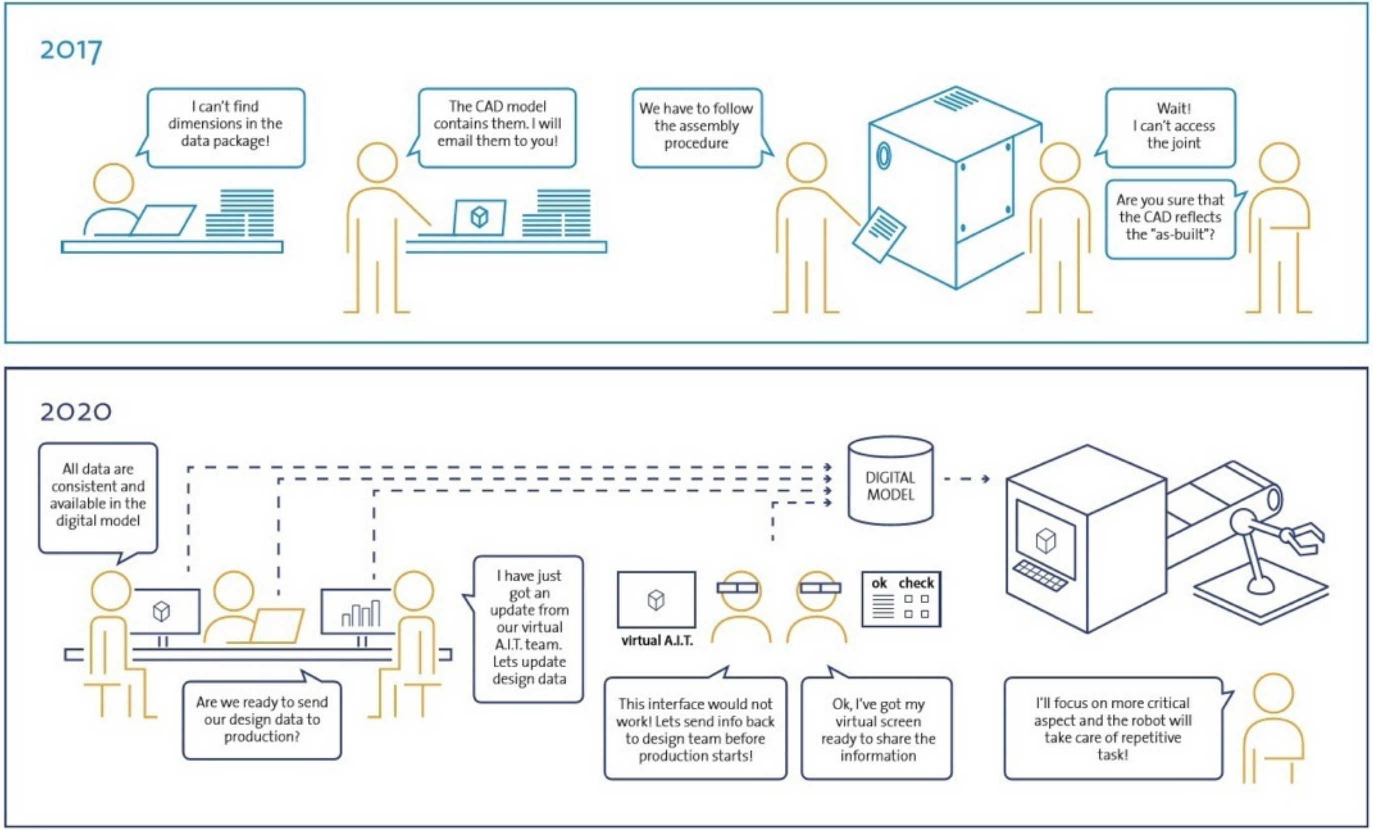



Source: SpaceX/C. Kuehmann

- Agile approach
- Model-Based Systems Engineering (MBSE)
  - OPM (Object Process Methodology), SysML (Systems Modeling Language), Modelica …
- Digital Twins, Cyber-Physical-System (CPS)

# Digitalization



# Theme 3 Summary

- Systems Engineering
  - Aerospace vehicles and other systems are becoming more complex and need at least 3-4 layers of decomposition (“magic” number 7)
  - “V”-Model of Systems Engineering is the classic approach
    - Starts with Stakeholder Analysis all the way to operations and Lifecycle Management
    - Importance of stage gates (‘milestones’): SRR, PDR, CDR, FRR, …
  - Several standards exists that codify how SE should be done (ECSS, NASA, INCOSE, ISO …)
- Requirements set constraints and goals
  - Essential for driving system/sub-system design
- Verification and Validation
  - Verification makes sure the product is built to requirements: Every requirement must be verified to ensure that the proposed design actually satisfies the requirement
  - Validation assesses whether the product/system is really what the customer wants, i.e. whether it satisfies his or her needs
- Traceability of all the Systems Engineering processes

# Next week



- Theme 4: Materials

- Exercise 2.2

- First part of Mini-project:
  - Part 1 – Requirement analysis. *Deadline: 20.03.25*
  - Part 2 – Architecture and potential components. *Deadline 10.04.25*
  - Part 3 – Concept. *Deadline 01.05.25*