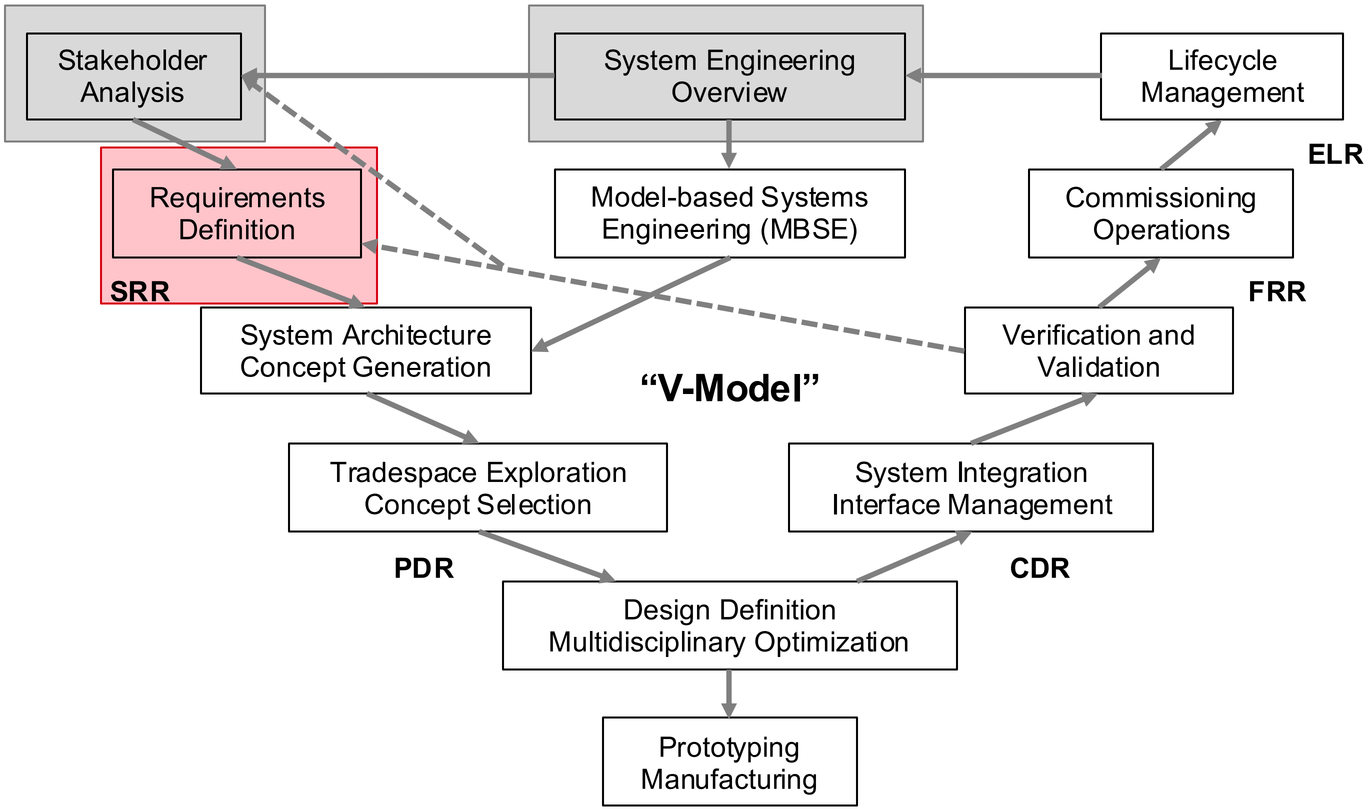
EE584 – S/C Design and Systems Engineering

Lecture 3
Requirements

Gilles Feusier & Volker Gass

Based on the lecture "Fundamentals of Systems Engineering"
By Prof. Olivier L. de Weck MIT/EPFL

The “V-Model” of Systems Engineering



EPFL

Overview

- **Requirements**
  - What are requirements?
    - Definition, Examples, Evolution, Standards
  - Challenges of Requirements Definition
    - Flowdown and Allocation
    - Writing good requirements
    - Validation and Verification
  - What happens at the SRR?
- **System Architecture and Concept Generation**
- **Concept Selection and Tradespace Exploration**

EPFL

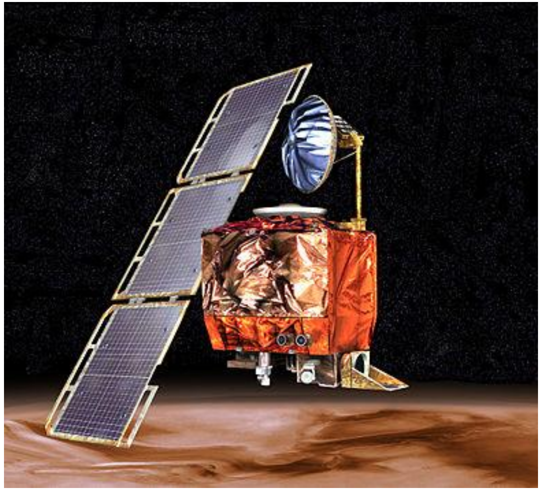
Requirements Definition

- 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**

EPFL

Poor requirements example: MCO

- Mars Climate Orbiter (MCO) was launched by NASA on December 11, 1998
- Intended to study Martian climate, weather and surface changes and act as communications relay back to Earth
- **However, disintegrated during orbit insertion on Sept 23, 1999 → approach too close → requirements not followed**
- Units confusion problem: Ground Software produced output in non-SI units (lbf-sec) instead of SI units: Ns
  - Calculation of total momentum produced by engine burns needed by Guidance, Navigation, and Control (GNC)
- Contract between NASA and Lockheed Martin **did** specify SI-units
  - This requirement was flowed down to the Software Interface Specification (SIS), but not verified later and not implemented in the file used for the propulsion maneuvers.



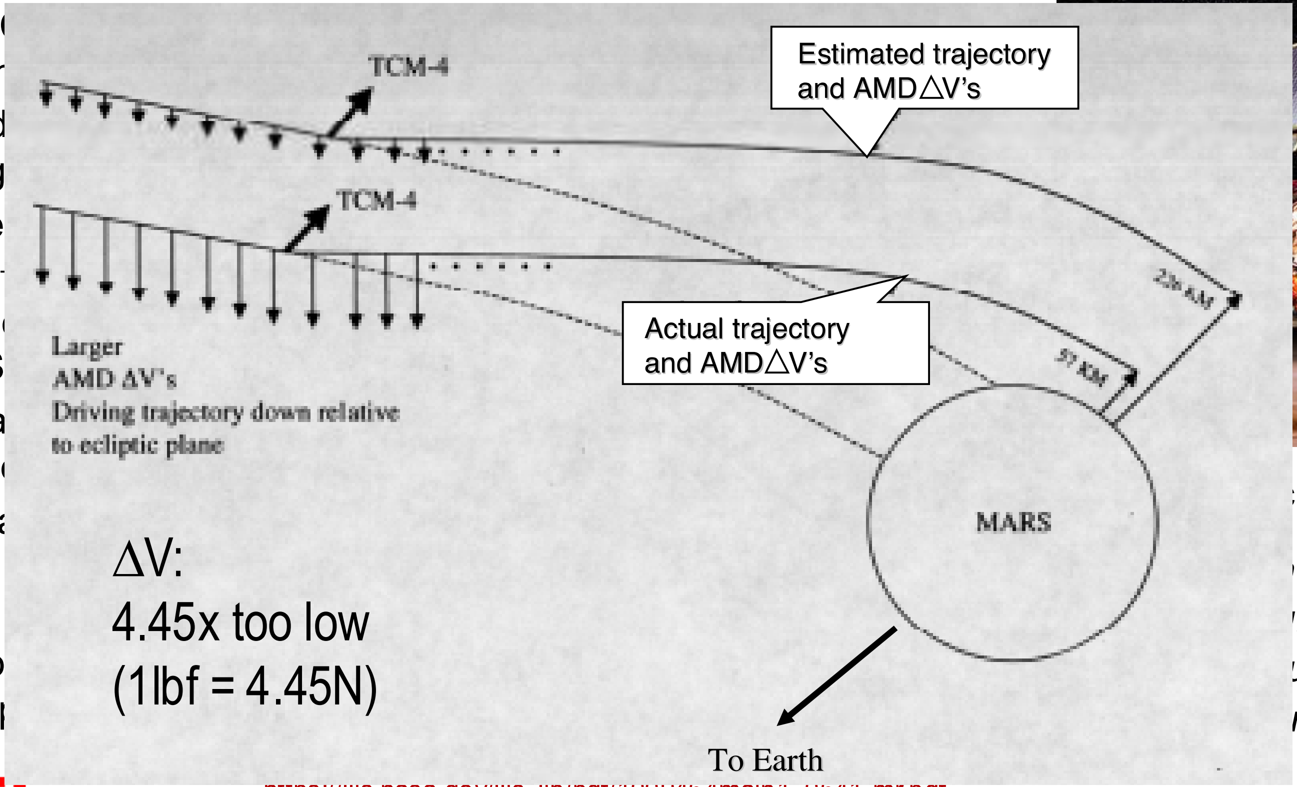
From the accident report:

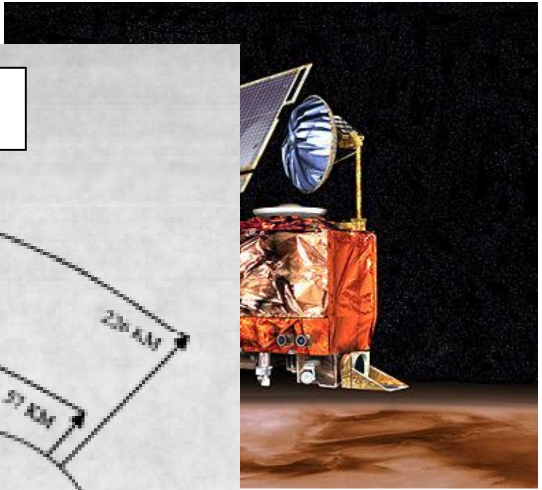
“Items that the mission assurance manager could have addressed for MCO included ensuring that the AMD file met the requirements of the SIS ..”

https://llis.nasa.gov/llis_lib/pdf/1009464main1_0641-mr.pdf

Poor requirements example: MCO

- Mars (
Decer
- Intend
chang
- **Howe**
1999 -
- Units
non-S
  - Ca
nec
- Contra
units
  - Th
Sp
im





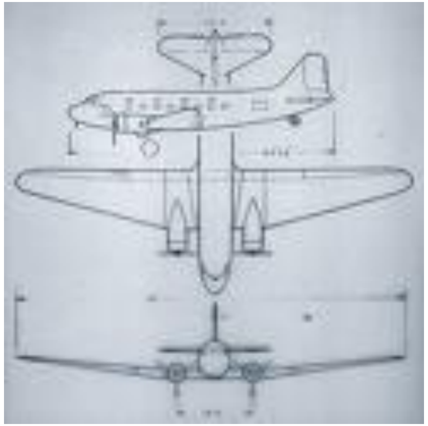
ident report:

e mission assurance
ld have addressed
uded ensuring that
met the requirements

https://llis.nasa.gov/llis_lib/pdf/1009464main1_0641-mr.pdf

Good requirements example: DC-3

- **Requirements based on desired improvements to DC-2**
- **Very simple**
  - 3 page RfP (McDonnell Museum)
  - “Marathon” phone call between Smith and Douglas
- **Key Requirements**
  - Range: 1000 miles
  - Cruise Speed: 150 mph
  - Passengers: 20-30
    - Depending on configuration
  - Twin Engines
  - “Rugged and Economical”



1st flight: 17 Dec 1935
Over 10,000 built

EPFL

Requirements Explosion since 1970s

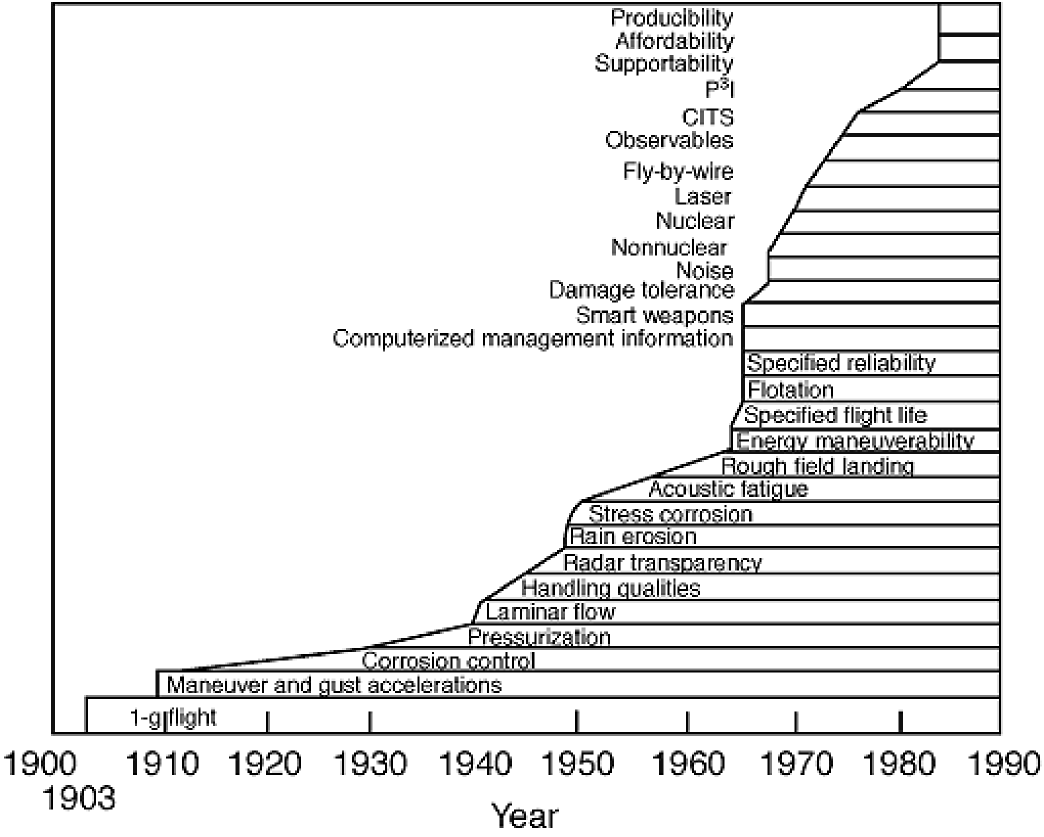


Fig.1 Design requirements growth for aerospace vehicles.

More and more requirements were added as systems grew in performance and complexity

Source:
AIAA MDO TC
White Paper, 1991

EPFL

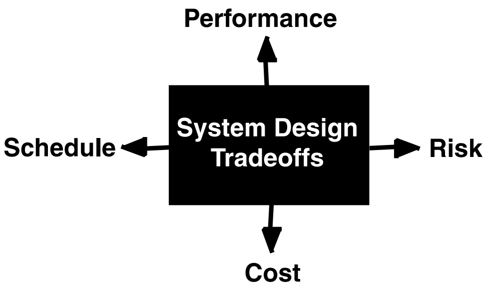
Requirements Standards and Handbooks

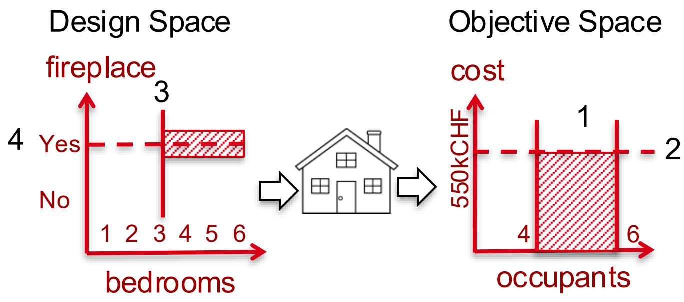
- **NASA Systems Engineering Handbook**
  - NASA/SP-2016-6105
    - Section 4.2 (pp. 63-72) – Technical Requirements Definition
    - Section 6.2 (pp. 150-155) – Requirements Management
    - Appendix C (pp. 236-239) – How to write a good Requirement
    - Appendix D (pp. 240-242) – Requirements Verification Matrix
- **International Council of Systems Engineering (INCOSE)**
  - Systems Engineering Handbook, Version 5
  - Requirements Working Group
    - https://www.incose.org/incose-member-resources/working-groups/analytic
- **ISO/IEC 15288 (IEEE STD 15288-2023)**
  - Systems and software engineering —
  - System life cycle processes
    - 6.4.2 Stakeholder needs and requirements definition process
- **ECSS Space Engineering - Technical requirements specification**
  - ECSS-E-ST-10-06C
    - Processes and definitions (ESA may sometimes differ from NASA)

EPFL

Requirements set *constraints* and *goals* in the design and objective space

- 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
- Two main spaces:
  - Design Space – the things we decide as engineers
  - Objective Space – the things our systems/products achieve and what our customers care about





1. "The house shall sleep between 4 and 6 people"
2. "The total build cost should be less than 550kCHF"
3. "The house shall have at least 3 bedrooms"
4. "The house should have a fireplace"

EPFL

Purpose of Technical Requirements Definition

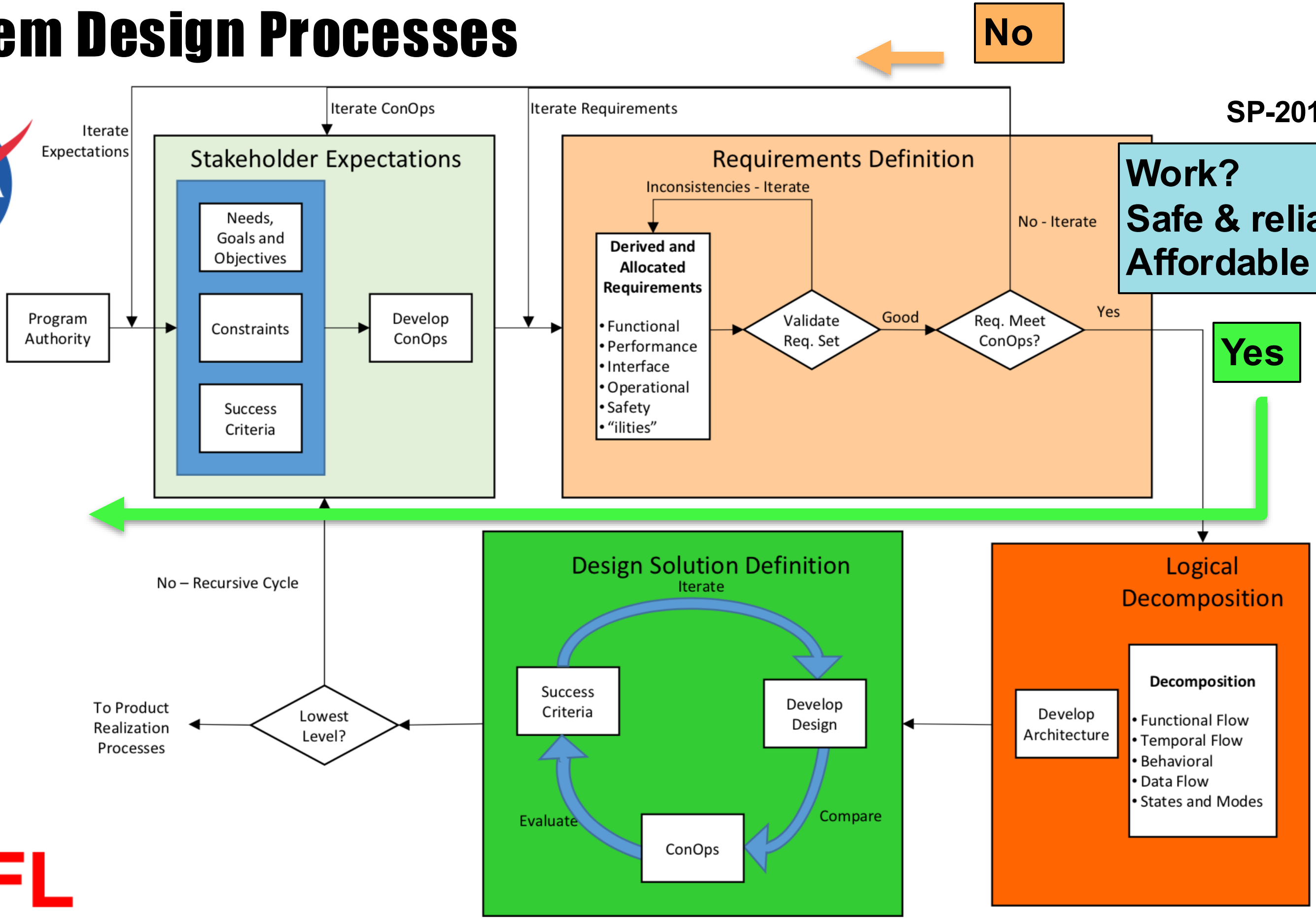
- **The Technical Requirements Definition Process**
  - Is used to transform the baselined stakeholder expectations (input) into unique, quantitative, and measurable technical requirements (output)
- **Requirements**
  - Come in many flavors
  - Should be expressed as well-written “shall” statements that can be used for defining a design solution

EPFL

Relationships among the upstream System Design Processes

SP-2016-6105, Figure 4.0-1

Work?
Safe & reliable?
Affordable (incl. Schedule)?



Types of Requirements

- **Functional Requirements define what functions need to be done to accomplish the mission objectives**
  - Example: The Thrust Vector Controller (TVC) shall provide vehicle control about the pitch and yaw axes.
    - This statement describes a high level function that the TVC must perform.
    - Statement has form of Actor – Action Verb – object acted on
- **Performance Requirements define how well the system needs to perform the functions**
  - Example: The TVC shall gimbal the engine a maximum of 9 degrees, +/- 0.1 degree
- **Constraints are requirements that cannot be traded off with respect to cost, schedule or performance**
  - Example: The TVC shall weigh less than 55 kg (mass, power, volume …).
- **Interface Requirements** ∃ Internal and External interfaces
  - Example: The TVC shall interface with the J-2X per conditions specified in the CxP 72262 Ares I US J-2X Interface Control Document, Section 3.4.3.
- **Environmental requirements**
  - Example: The TVC shall use the vibroacoustic and shock [loads] defined in CxP 72169, Ares 1 Systems Vibroacoustic and Shock Environments Data Book in all design, analysis and testing activities.
- **Other -illities requirement types described in the SE Handbook include: human factors, reliability requirements, and safety requirements.**

EPFL

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

Requirements Writing “Workshop”

- **Turn to your partner exercise**
- Together write a good requirement that was (possibly) used in the development of one of the following solutions
  - A – Mr. Sticky tape for trapping flies
  - B – New BMW iX2 electric car
  - C – EPFL Rolex Center

A

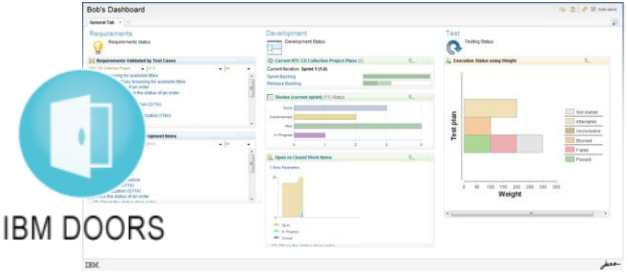
B

Source: BMW AG

C

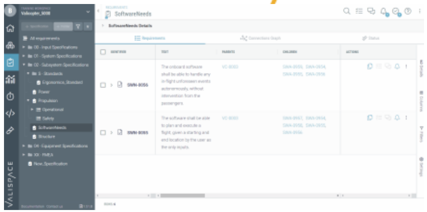
Requirements Capture: Documents vs. Database

- **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)



https://www.ibm.com/fr-fr/products/requirements-management

VALISPACE



Valispace
https://www.valispace.com

ReqView
https://www.reqview.com

EPFL

Overview

- **Requirements**
  - What are requirements?
    - Definition, Examples, Evolution, Standards
  - Challenges of Requirements Definition
    - Flowdown and Allocation
    - Writing good requirements
    - Validation and Verification
  - What happens at the SRR?
- **System Architecture and Concept Generation**
- **Concept Selection and Tradespace Exploration**

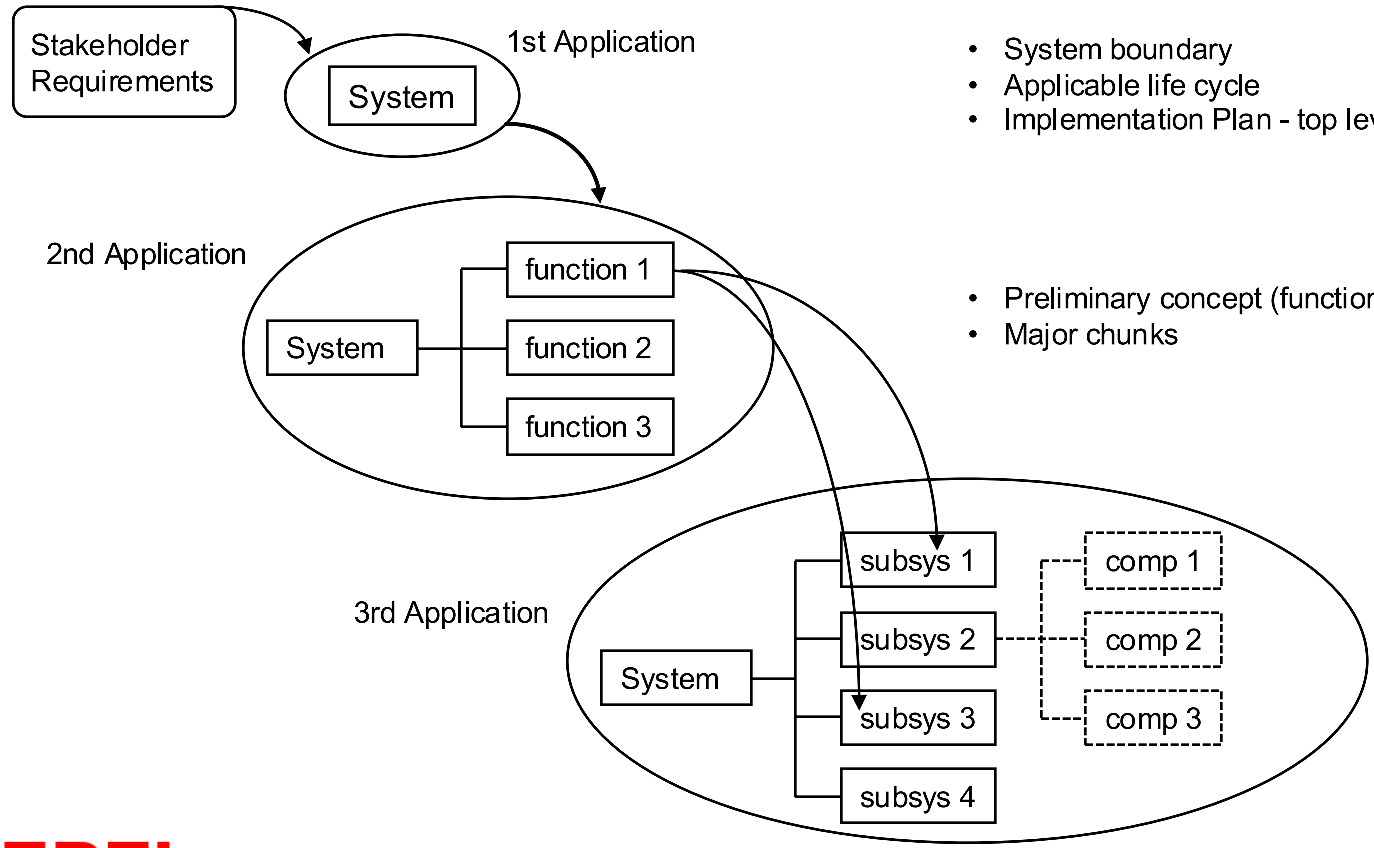
EPFL

Requirements Allocation

- **Decompose system requirements into lower levels of design.**
  - Define all the lower level functions which must be performed to satisfy the requirement
  - Create architecture of sub-components to provide those functions
- **Allocate a level of performance to each lower level function**
  - Specify interface requirements to other sub-systems
- **Closure - Ensure that satisfaction of the set of requirements at the lower level will guarantee satisfaction of the higher level requirement.**
- **Keep Traceability** (why, modifications, revisions)

EPFL

Requirements Allocation Process



- System boundary
- Applicable life cycle
- Implementation Plan - top level

- Preliminary concept (function to form)
- Major chunks

- System key performance parameters
- Subsystems defined
- Notional components

Difficult to decompose requirements to lower levels while staying solution neutral

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**

EPFL

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

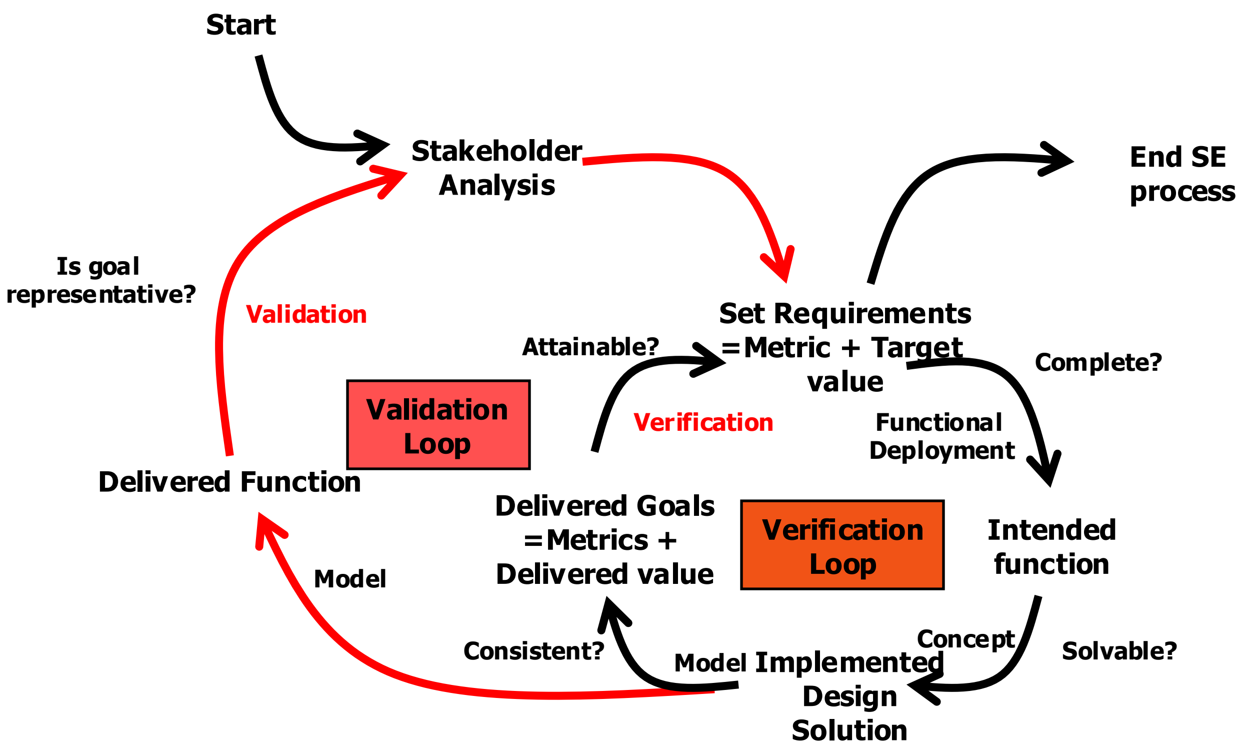
- **Requirements only written for "first use"**
- **Over-specifying**

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



Overview

- **Requirements**
  - What are requirements?
    - Definition, Examples, Evolution, Standards
  - NASA Requirements Process
  - Challenges of Requirements Definition
    - Writing good requirements
    - Flowdown and Allocation
    - Validation and Verification
  - What happens at the SRR?
- **System Architecture and Concept Generation**
- **Concept Selection and Tradespace Exploration**

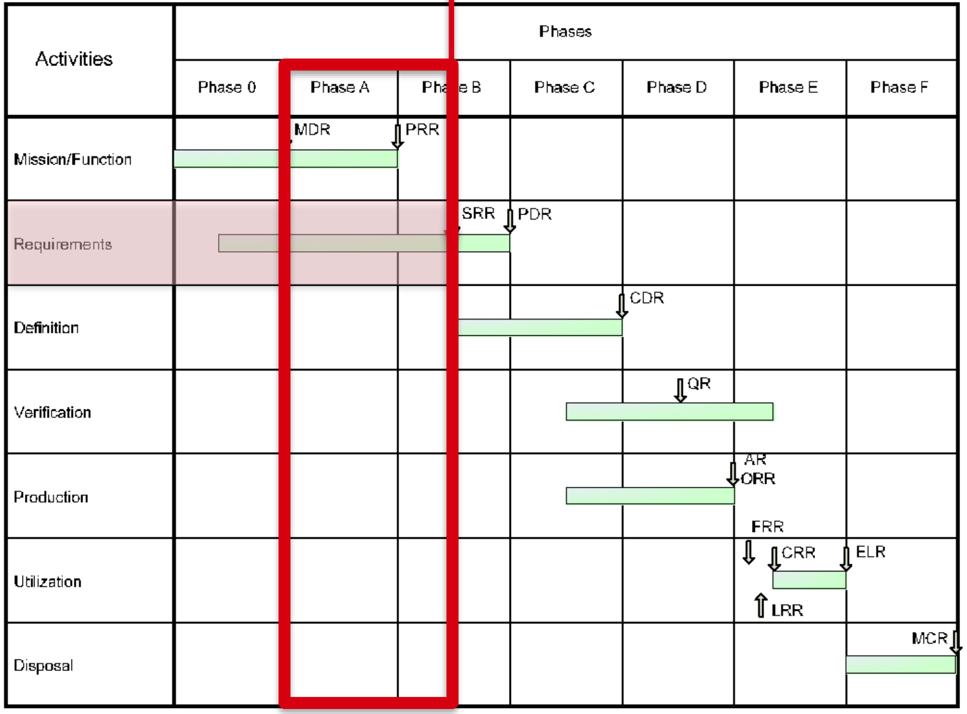
EPFL

The life cycle of space projects - ESA

SRR

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

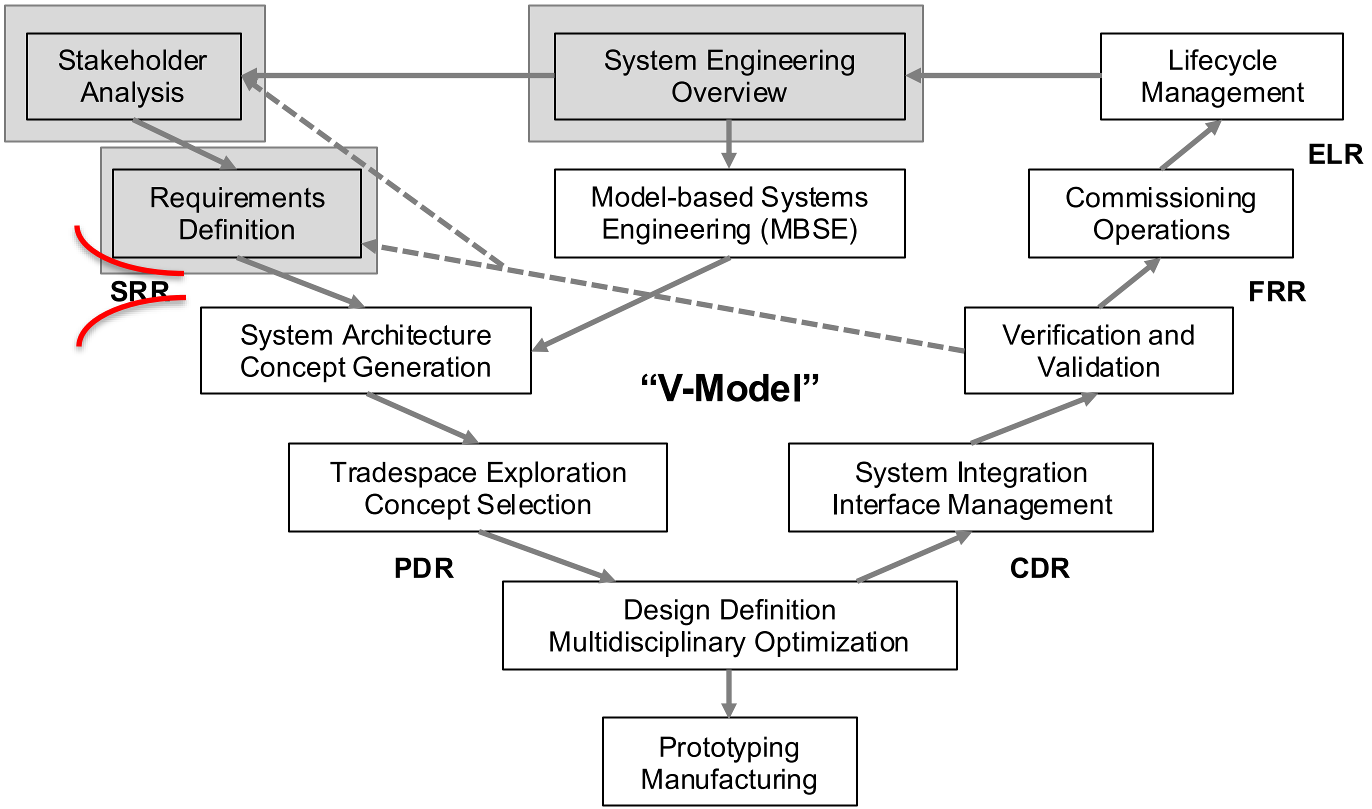
- Phase 0 - Mission analysis/needs identification
- **Phase A - Feasibility**
- **Phase B - Preliminary Definition**
- Phase C - Detailed Definition
- Phase D - Qualification and Production
- Phase E - Utilization
- Phase F - Disposal



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

EPFL

The “V-Model” of Systems Engineering



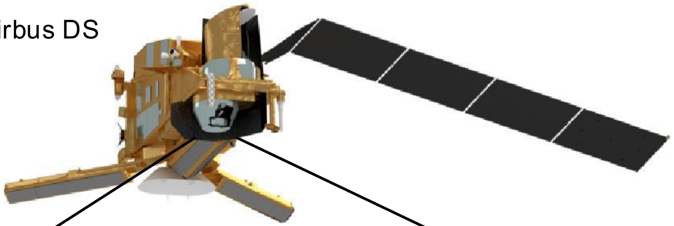
EPFL

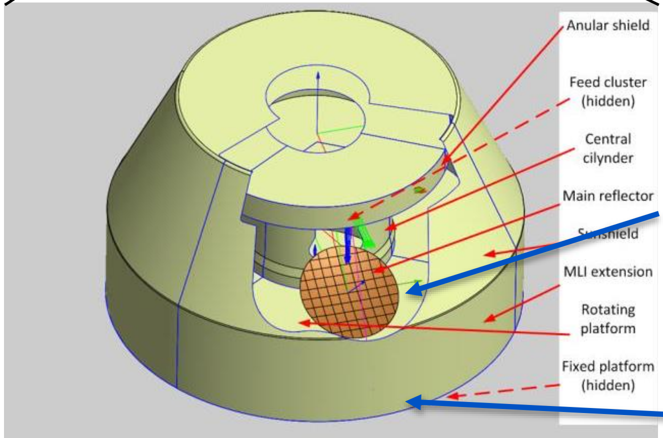
System Requirements Review (SRR)

- **SRR is primarily a human / social peer-review process**
- **Main goal of SRR: Vet the requirements as written. Find any missing, misstated, redundant or otherwise unsatisfactory requirements.**

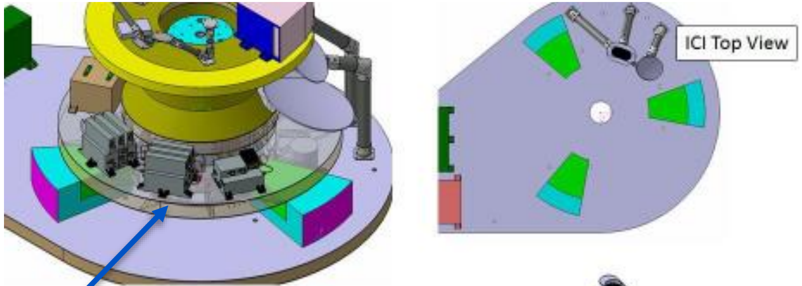
Ice Cloud Imager Launch Lock Device (MetOp-SG)

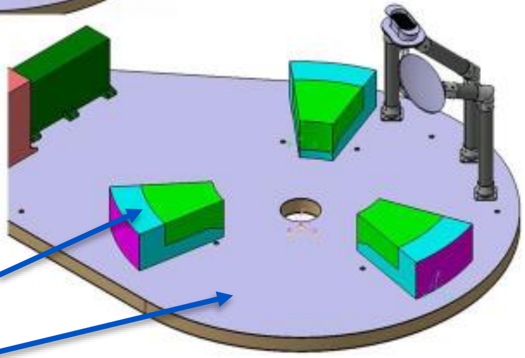
© Airbus DS





© D. Marote et al., 39th ESA Antenna Workshop 2017





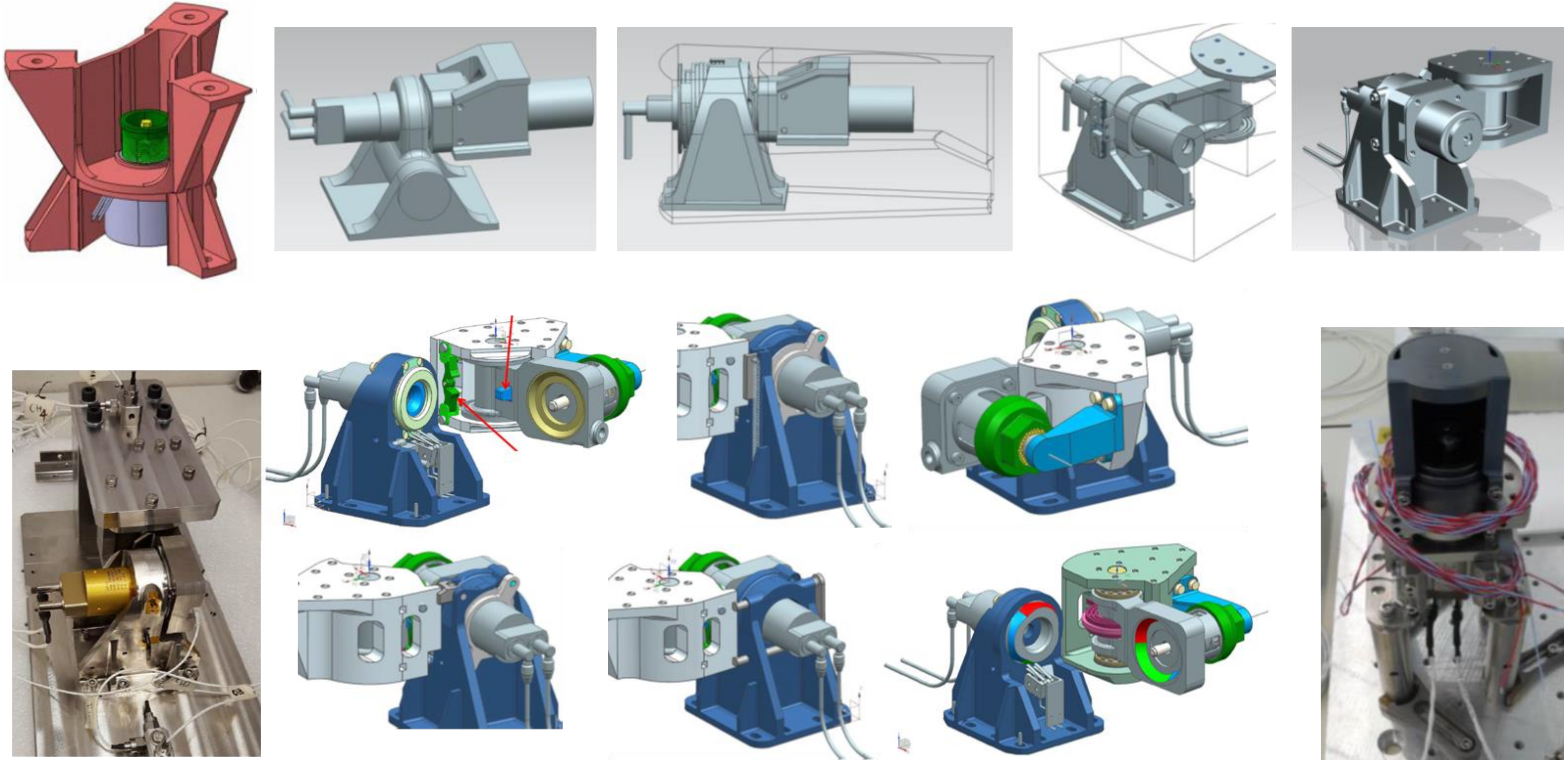
Rotating platform

LLD envelope

Fixed platform

© G. Smet et al., 18 ESMATS 2019

Ice Cloud Imager LLD - Iterations



© G. Smet et al., 18 ESMATS 2019

Requirements Definition - Summary

- **Good Requirements are essential for driving system design**
  - Basis for cost and schedule estimation
  - Sets up necessary verification and validation activities
- **Requirements flow-down is challenging**
  - Level 0 / Level 1 Requirements from Stakeholder analysis and ConOps
  - Level 2, 3 etc… requirements emerge later during Preliminary Design
- **Some methods and also commercial tools exist for formal requirements management**
  - Isoperformance → requirements allocation given upper level targets
  - DOORS → part of IBM Rational Suite. Professional requirements management