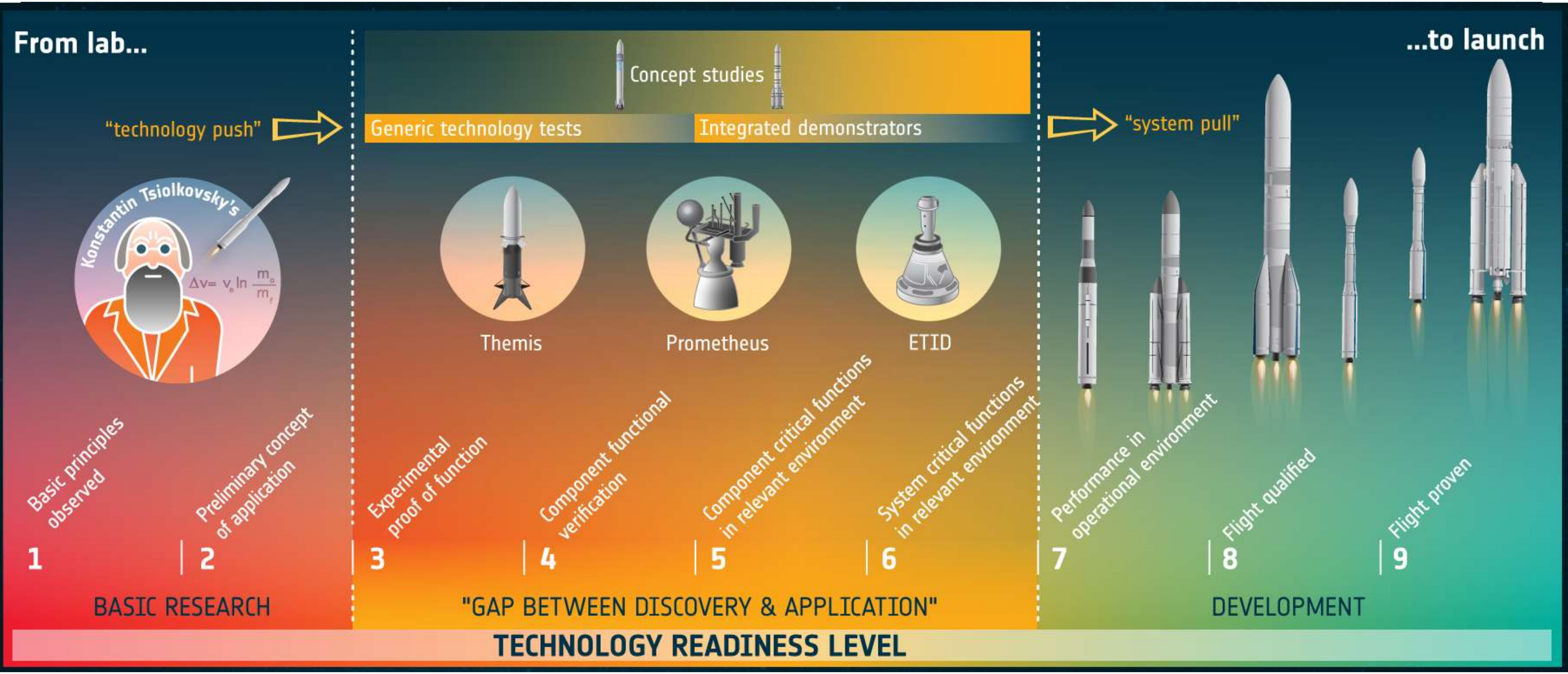
Technology Readiness Levels

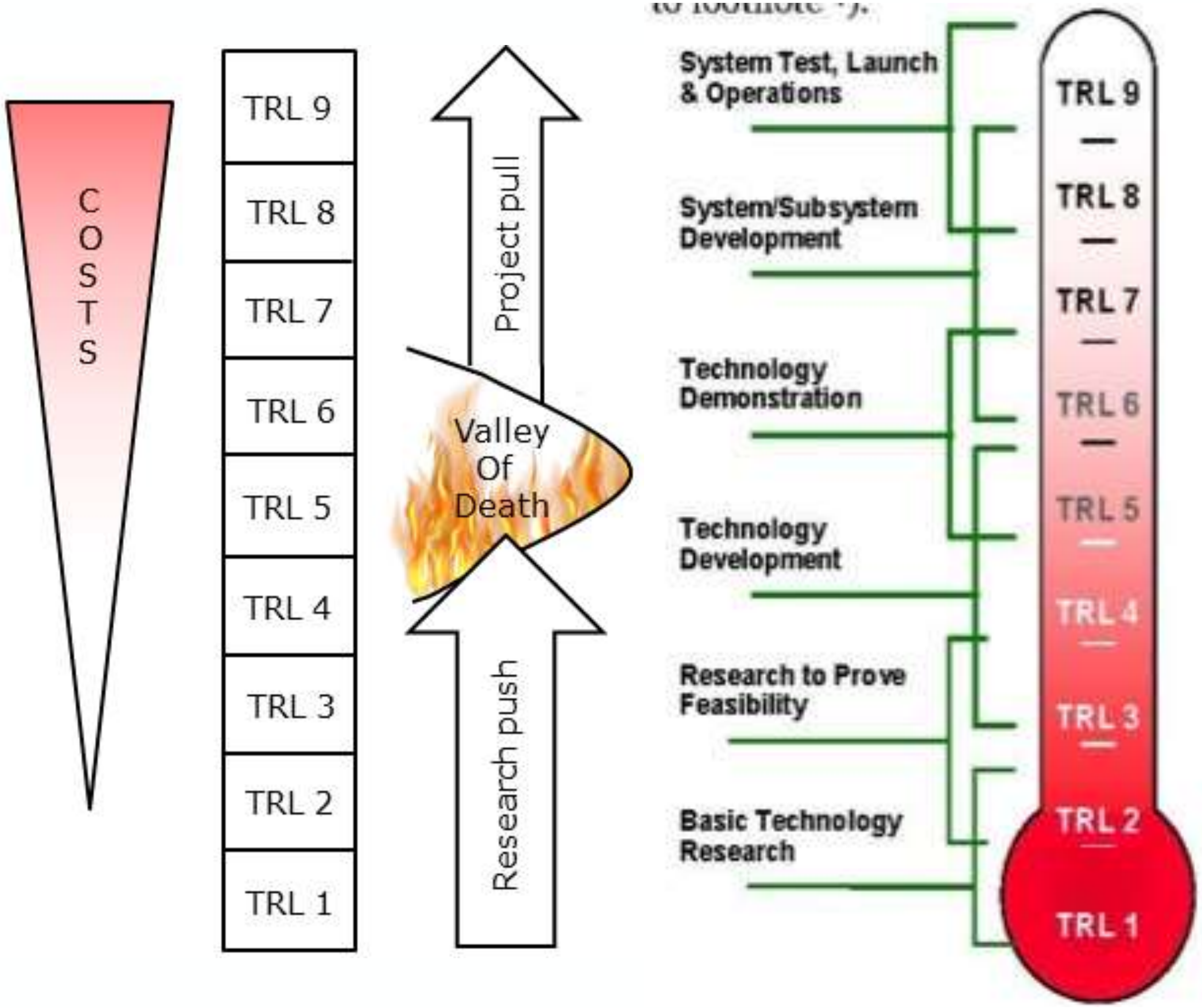
Technology readiness level (TRL) guidelines

Reference: ECSS-E-HB-11A



Technology readiness level (TRL)

ECSS-E-HB-11A – Technology readiness level (TRL) guidelines



Technology Readiness Levels	
9	Actual system "flight proven" through successful mission operations
8	Actual system completed and accepted for flight ("flight qualified")
7	Model demonstrating the element performance for the operational environment
6	Model demonstrating the critical functions of the element in a relevant environment
5	Component and/or breadboard critical function verification in relevant environment
4	Component and/or breadboard functional verification in laboratory environment
3	Analytical and experimental critical function and/or characteristic proof-of-concept
2	Technology concept and/or application formulated
1	Basic principles observed and reported

How to write good technical requirements for a product development

(or: How to define what is expected before starting to design something.)

Disclaimer

- This training is intended as a basic introduction to technical requirements for those struggling to understand what is expected.
- It is intended to help you write and monitor your requirements for technology development and product development activities (and proposals).
- It does not cover Mission level requirements nor requirement engineering nor requirement engineering tools.

Contents

- Requirements
  - Why we need requirements
  - Definition of requirements
  - Types of requirements
  - Requirement detail level vs TRL
  - Requirements / recommendations for formulating technical requirements
- Managing and controlling requirements
- Some “golden rules”…

Requirements

What is a requirement?

A requirement is a statement that captures the understanding of:

- **What** the end product has to do (→ functional requirements)
- **How well** the end product has to do it (→ performance requirements)
- Under what **constraints** the end product has to perform these functions (→ constraint requirements e.g. physical, environmental etc.)

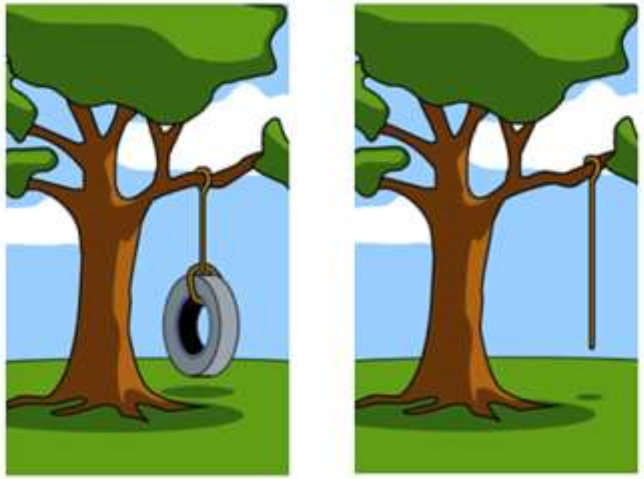
Why we need requirements

Requirements ensure that the end design/product is fit for purpose.

Requirements should form the reference for all your design work, analyses, verification and testing.

Developing with no or incomplete requirements is like driving a car with no map and no destination!...

What the customer wanted…



What was delivered due to poor requirements specification

Why we need requirements

Good requirements will:

- Establish a basis for agreement between the customers/stakeholders and the development team.
- Reduce development effort by avoiding work going in the wrong direction.
- Provide a strategy for the verification programme.
- Ensure the item will perform its desired goal / objective and fit with any other systems
- Provide a metric to resolve technical design trade offs
- Serve as a basis for promoting the finished product.
- Enable a clear identification of changes (and hence who pays).
- Enable different groups to simultaneously work on different parts of a larger system.

Mission and Product Requirements

Mission Requirements:

- Defined based on the end user needs, and iterated between customer and supplier until a solid baseline is established
- Expanded on and flowed down to systems then sub-systems and eventually lower tier suppliers by the prime: All lower level requirements derive from and contribute to higher level requirements.
- Requirement tracking is critical through all levels to ensure Mission requirements are fulfilled.

Product/ Service Development Requirements: (most relevant for RPA / PECS calls)

- Established by the company intending to develop the product/ service – taking customer needs into account
- Should take into account the market/ competitor products to ensure a successful product.

This presentation focusses on the Product/Service Development Requirements

Mission and Product Requirements

Mission

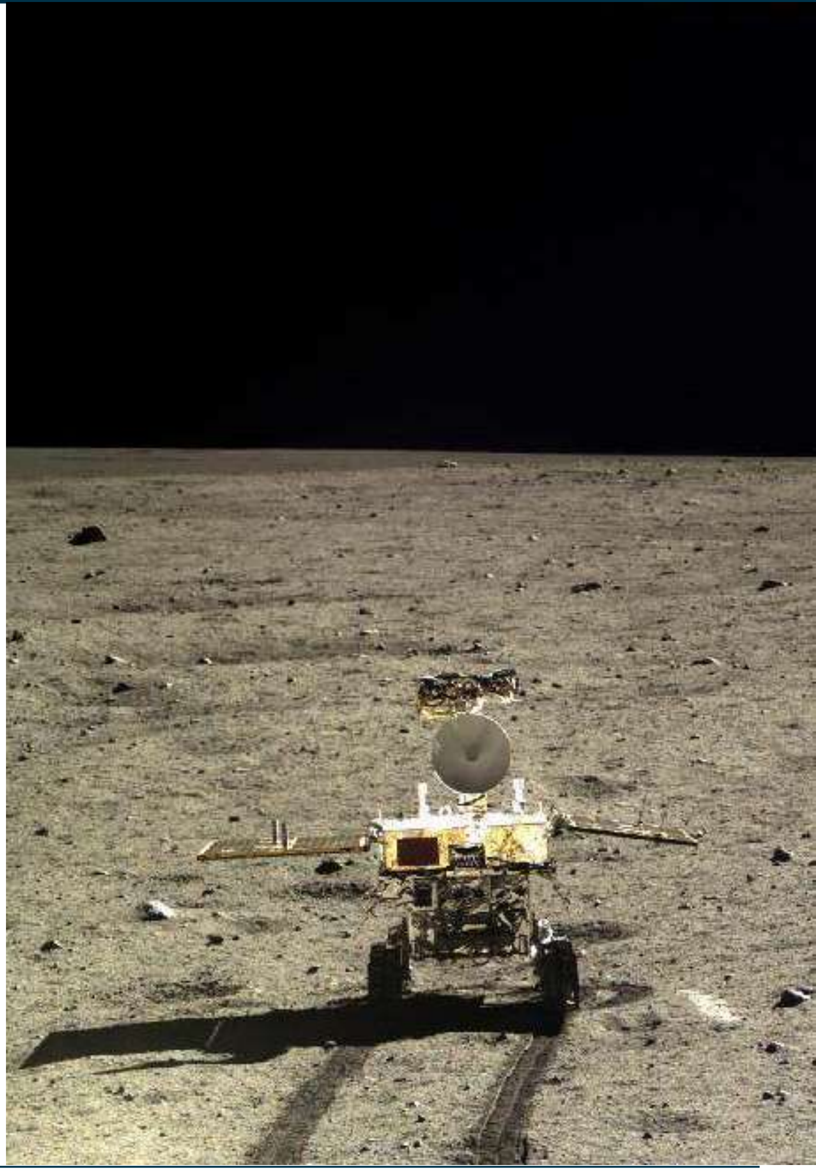
to collect and bring back lunar samples

Rocket

Rover

Total mass of rover shall not exceed 500kg.

Solar panel shall not weigh more than 10kg.



What the ECSS says about requirements

ECSS-E-ST-10-06C “Technical requirements specification”

- Provides “requirements on requirements” –i.e. how to:
  - Identify and capture requirements
  - Write requirements
    - including how to formulate “good” unambiguous requirements
    - definitions of “shall”, “should”, “may”, words to avoid...
  - Classify according to type of requirement
    - Functional, operational, physical, design…
- Used to write e.g. the Prime’s System Specifications
- It does not provide practical instructions on how to do requirements management.

This is needed for a mission definition and requirements flow down but is only informative for product/service development requirements under a call for proposals

Example of flown down requirement structure

Anatomy of a Technical Requirement as part of a mission / complex system

Heading

SA-123

/Created/

/ T,R,A

- Heading — brings the structure and adds searchability and readability
- SA-123 — unique requirement number needed for tracking
- /Created/ — context / version history (part of tracking tool)
- / T,R,A — verification methods (e.g. test, analysis, review of design, inspection as per ECSS-E-ST-10-02C)

This is overkill for a stand alone product / service development but when supplying to large projects it is typical of the format you will see

Types of requirements

Main types:

- Functional requirements: What does it have to do?
- Performance requirements: How well does it have to do it, e.g. accuracy, speed..?
- Physical requirements: How big and heavy can it be?
- Interface requirements: What does it have to connect to and how (h/w, s/w, MMI)?
- Environmental requirements: Under what conditions does it have to work?

All these are needed to be able to design any product / service!

Supplementary/ specific requirements:

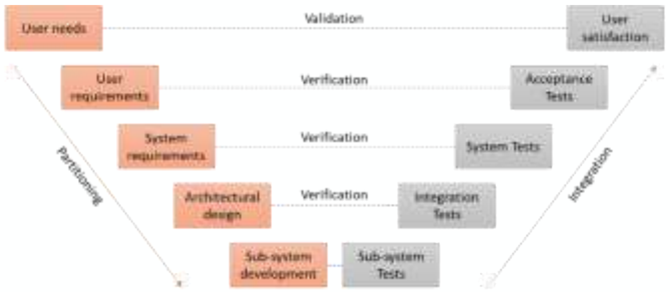
Mission requirements
Design requirements
Verification requirements
Cost requirements
Product Assurance (PA) induced requirements
Operational requirements
Human factor requirements

Requirement Evolution vs. TRL

TRL 1-2	TRL 3-4	TRL 5-6	TRL>6
First iteration of **key** driving requirement(s).	First iteration of **all** key requirement(s) and constraints. Some may still be TBD or TBC.	**Complete** set of **all** requirements defining and driving the product and their validation and qualification.	Complete **consolidated and frozen** set of requirements.
Overall goal (draft) Key Functional Key Performance	Mission requirement/goal (final) Functional Performance Physical Environmental Interface (driving only)	Mission requirement/goal Functional Performance Physical Environmental Interfaces (all) Operational Human factor requirements Product Assurance (PA) induced requirements Design requirements (i.e. design constraints) Verification requirements GSE Requirements (draft)	Mission requirement/goal Functional Performance Physical Environmental Interface Operational requirements Human factor requirements Product Assurance (PA) induced requirements Design requirements (i.e. design constraints) Verification requirements GSE requirements

Requirement Formulation: Key aspects

- **Each Requirement Statement shall be:**
  - **Clear and unambiguous** – readily understandable and only one interpretation is possible.
  - **Correct and consistent** – does not contain error of fact or contradiction to another requirement
  - **Realistic** – can be satisfied within natural physical laws, state of the art technologies, and other project constraints
  - **Non-prescriptive** – Shall state what is to be achieved, not how it is to be achieved.
  - **Verifiable** – can be proven and demonstrated to have been achieved (e.g. through Inspection, Analysis or Test)



- **Organize requirements logically.**

- **Requirements shall be traceable.**
  - i.e. it shall be clear why the requirement is there and from where it was derived

Requirement Formulation: Word Use

Specific formulation for requirements:

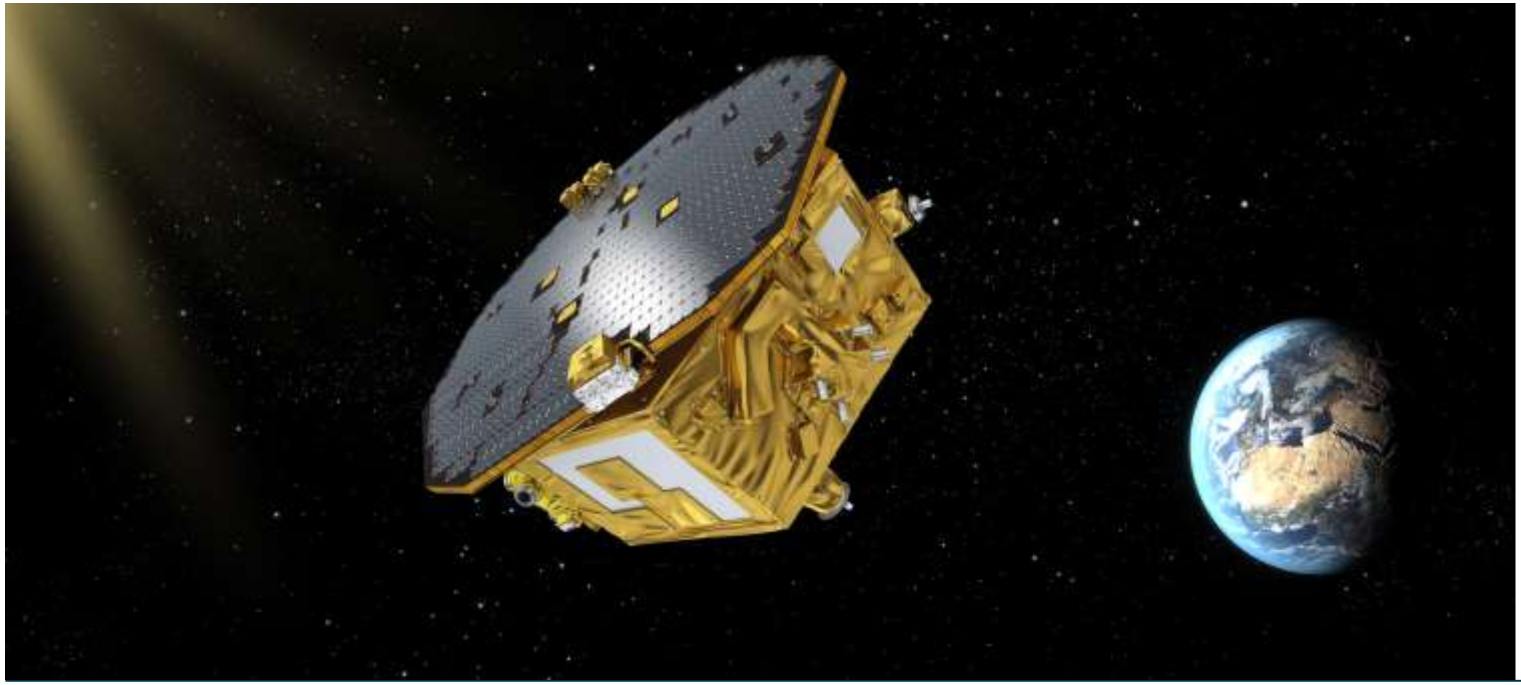
- Shall = a requirement that must be met
- Will = facts or declaration of purpose
- Should = a goal
- Could = an option

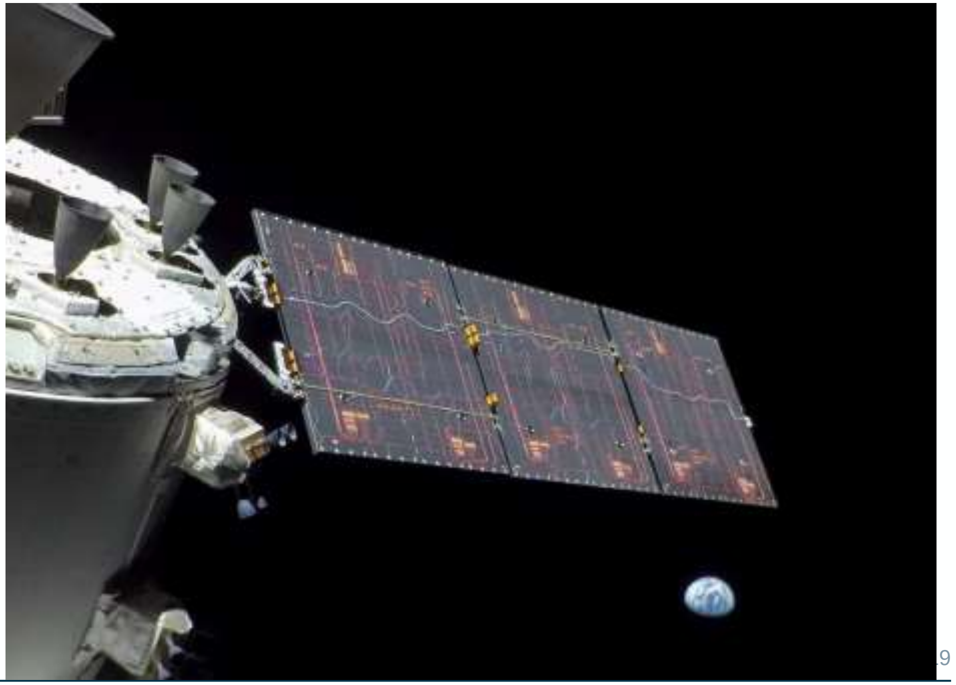
Example:

- The rocket shall launch to the Moon.
- The rocket will be used to bring humans to the Moon.
- The rocket should be reusable.
- The rocket could use liquid or solid propellants.

Example Product Requirements

- The system shall operate at a power level of...
- The software shall acquire data from the...
- The structure shall withstand loads of...
- The hardware shall have a mass not exceeding...





Example 1: Crop identification tool using EO data

The service shall monitor crop conditions and forecast yield using ESA Earth Observation (EO) data.

- The service shall operate on both Windows and Mac machines.
- The service shall use data from the Landsat and Sentinel-2A satellites.
- The service shall display an area of $1km^2$.
- The service shall have a crop monitoring resolution of 5m.
- The service shall forecast crop yield with a margin not exceeding 5%.
- The service shall have a running cost not exceeding 25k€ 7 year.

Example 2: Solar Array (adapted from a real ESA mission)

Main functional requirements:

The Solar Array (SA) shall provide the following main functions:

- Conversion of solar radiation into electrical power by means of solar cells
- Provision of electrical power to the spacecraft during sunlit period
- Provision of all necessary electrical wiring and protection between the cells and the electrical interface to the spacecraft
- Mechanical support of the electrical elements
- Thermal control of the assembly and sunshielding of the spacecraft
- A solar flux monitor

Example 2: Solar Array (adapted from a real ESA mission)

The main functional requirement is broken further down into:

SA-01/Derived from SRS-1701/A,R

The Solar Array shall provide at least 650W at the Solar Array / spacecraft interface connector in the following conditions:

- 35V at the Solar Array / spacecraft interface
- Sun off pointing by up to 5 degrees to the panel
- One string failed
- Degradation and losses as defined in SA-00

Example 2: Solar Array (adapted from a real ESA mission)

Environmental requirements:

Lifetime / Ageing:

SA-48/Derived from SRS-2001/A,R

The Solar Array shall be designed for a minimum lifetime of:

- 60 months on ground (including testing and storage time)
- 24 months in orbit

During the possible total period, all spacecraft requirements shall be met.

SA-46/Derived from SRS-1701/A,R

The Solar Array supplier shall perform the design and analyses to establish the calculated temperature range that the Solar Array will experience throughout the mission using the environment shown in table ABCD.

Example 2: Solar Array (adapted from a real ESA mission)

Verification requirements:

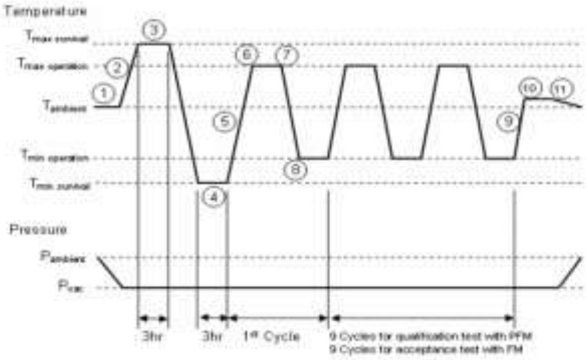
Thermal / vacuum test requirements:

The supplier shall demonstrate by test, compliance to the thermal and pressure environment requirements specified below:

SA-29/Created/T

For Test Sample Level Thermal Vacuum Tests, test levels for the TAT/DVT and PST shall be derived from the thermal environment defined in section 4.2 with the qualification margin included. The test sequence defined in table EFGH shall be followed.

	TAT/DVT/PST Test Temperature Level
Initial 3 Hour Hot Soak	Max Survival Temperature + qual margin
3 Hour Cold Soak	Min Survival Temperature - qual margin
Upper Test Temperature	Max Operating Temperature + qual margin
Lower Test Temperature	Min Operating Temperature - qual margin
Number of Test cycles	500 (TBC)



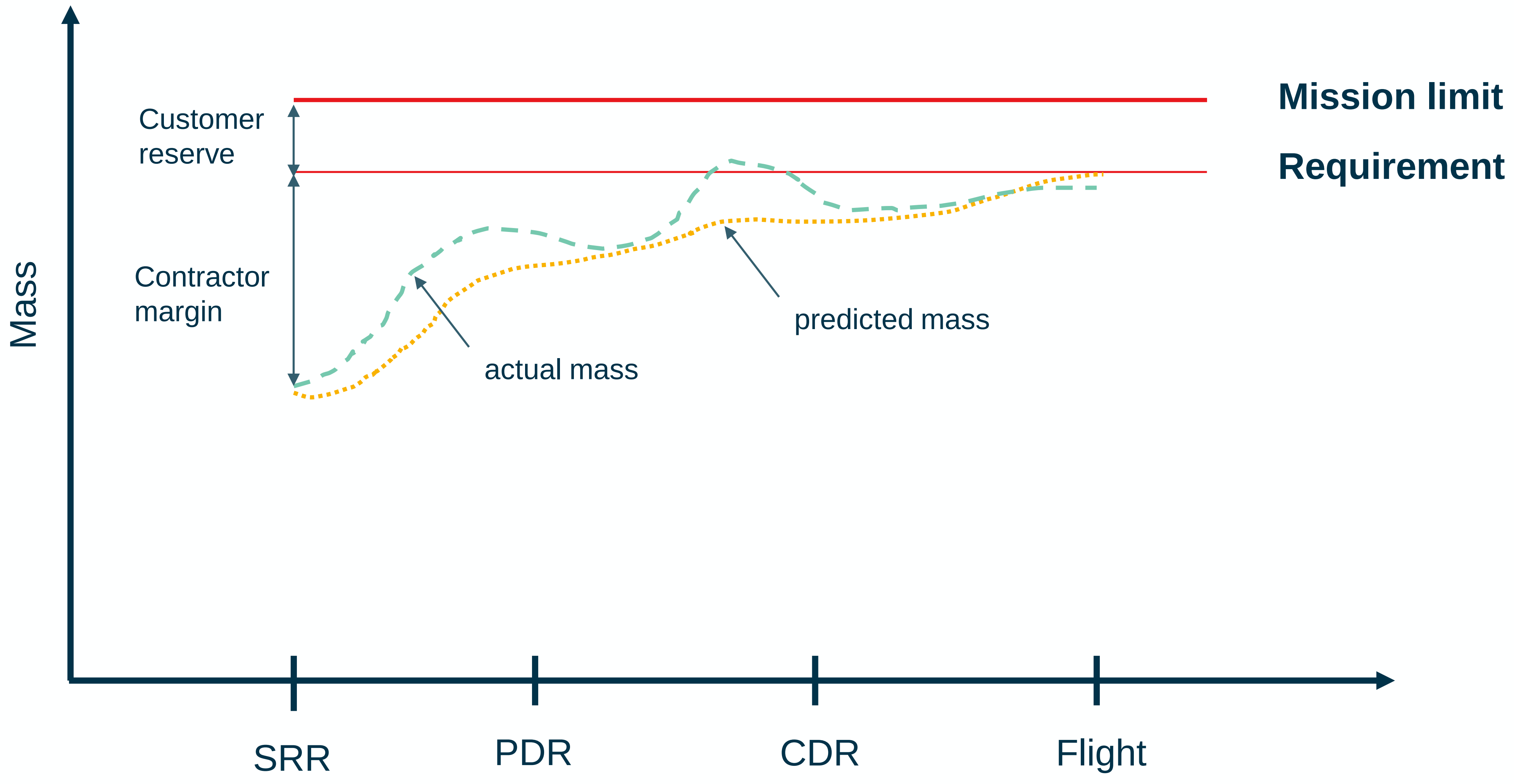
Managing and controlling requirements

Requirements Margins Management

Definition "margin":
The allowances carried in budget, projected schedules, and technical performance parameters (e.g., mass, power, memory, …) to account for uncertainties and risks during early design and analysis.

- margins evolve during the development of the mission, and they decrease with the increase in knowledge of the system
- margins shall not change the requirements, they change towards target design value, typical guidelines:
  - Pre-SRR: 30% margin
  - pre-PDR: 20% margin
  - pre-CDR: 10% margin
  - Acceptance Review (*): 5% margin

Requirements Margins Management



Verification of requirements

Definition Verification:

To establish confidence that the product / service will perform as expected.

- 4 fundamental methods for verifying a requirement:
  - Inspection
  - Analysis
  - Review of Design
  - Test
- The Verification Control Document (VCD) helps to systematically monitor and check the level of compliance with the requirements and follows the verification of those requirements through the design and test life cycle and is therefore a key tool for project management and risk control.

Verification Control Document

Example of a simplified Verification Control Document (VCD) and its evolution throughout the development process

Req. #	Requirement	Expected value	Compliance	Compliance notes	Analysis/ Inspection Reference	Test Plan Reference	Test Report Reference
Phy1	The mass shall be less than 2.0Kg	1.95Kg	C	Expected value includes 30% maturity margin	Design Report Issue 1.0, Table 2.4 - Mass budget		
Phy1	The mass shall be less than 2.0Kg	1.99Kg	C	Expected value includes 20% maturity margin	Design Report Issue 2.0, Table 2.4 - Mass budget	Physical Test Plan Issue 1.0, Section 3.0	
Phy1	The mass shall be less than 2.0Kg	2.01Kg	PC	Measured Value, request for deviation submitted, see NCR 001	Design Report Issue 2.0, Table 2.4 - Mass budget	Physical Test Plan Issue 1.0, Section 3.0	Physical Test Report Issue 1.0, Section 3.1

Each of the above lines corresponds to a project review, i.e. the VCD shall be presented and updated at each review for each requirement!

Impact of poor requirements

Poor requirements will lead to:

- non-functional, incompatible, non-competitive products or services
- creep in budget, timeline and resources
- re-design, re-work
- loss of (business) opportunities

Poor requirements - examples

- “The system must have good usability”, “easy to use”, “robust”
  → difficult / impossible to validate
- “Response time should be less than X seconds”
  → ambiguous, as it is open to different interpretation by different people leading to different results
- “The system has to be bug-free.”
  → difficult / impossible to meet / guarantee
- “We need to procure XY materials.”
  → this is a pre-requisite, not a technical requirement
- “We will do a market research.”
  → this is programme of work / implementation, not a technical requirement.

Some "golden rules"…

- Remember what requirements are for!
- Use the correct terminology (shall).
- Ensure requirements are complete (covering all areas), justified and verifiable.
- Refrain from designing the system or describe the planned flow of work.
- The use of "To Be Confirmed / Determined" (TBC / TBD) values should be minimized. TBC values shall be progressively reduced through the lifecycle of the development activity.
- Requirement compliance should be progressively monitored, validated and verified through the design life cycle.

The importance properly defined, traceable requirements

Example:

ITT for painting new UK Coast Guard Maritime Patrol Aircraft

Req 1 The aircraft shall be painted in a red and white stripes pattern ✓

Req 2 The stripe pattern shall be designed to ensure equal areas of red and white colour coverage ✓

Req 3 The stripe pattern shall consist of two diagonal red stripes and two diagonal white stripes ✗

esa

www.esa.int