# Systems Engineering and Aerospace Design (AE3211-I) 2022/2023

Dr. Fabrizio Oliviero, FPP (Course Responsible)
Prof. Dr. Eberhard Gill, SSE
Dr. Botchu Vara Siva Jyoti, SSE
Dr. Angelo Cervone, SSE

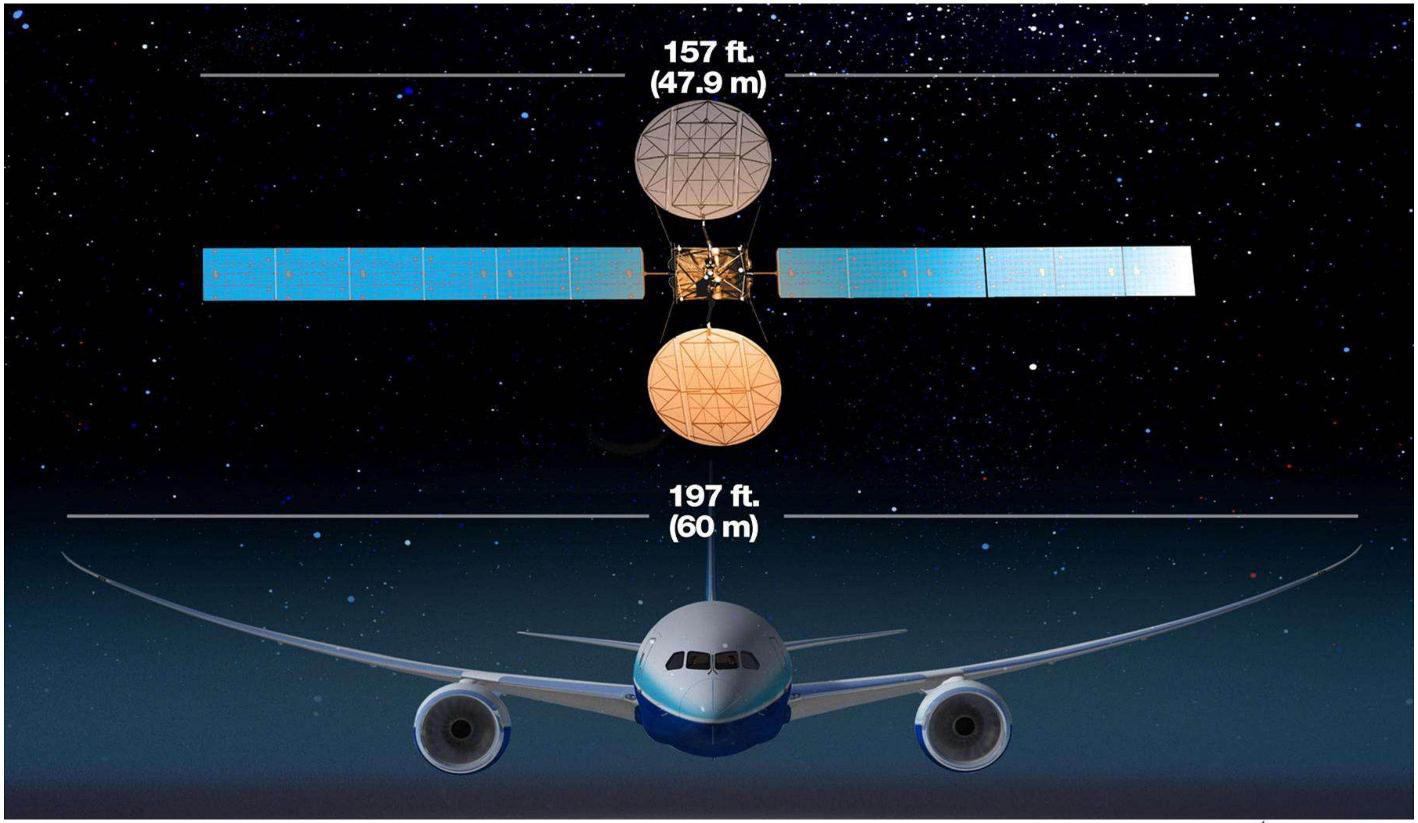
# Course Organization & Systems Engineering for Aerospace

# Structure

- Course Organization
  - Objective and Motivation
  - Format and Content
  - Schedule and Assessment

- Systems Engineering for Aerospace



# High-Level Learning Objective

After this course, you will be able to…

… Use Systems Engineering for engineering of complex aerospace products according to customer needs.

Required prior knowledge:

- Standard high school knowledge
- First B.Sc. course year completed
- AE2111-I and AE2111-II completed

  TUDelft

# Motivation

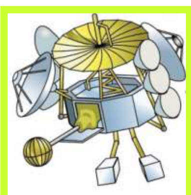
- Complete what you need to design spacecraft and aircraft systems!
- Understand how the elements from previous courses fit into a coherent framework of how to engineer a complex aerospace product!
- Appreciate using Systems Engineering also for non-aerospace products!
- Get prepared for the Design Synthesis Exercise!

This course is **horizontally** (i.e. timewise) **integrated** in the BSc curriculum (e.g. AE1111-I, AE1222-I, AE2111 II) and **vertically integrated** in the semester theme "Verification and Validation".
This course is part of the Module Aerospace Systems Engineering, Design and Production (AE3211).

 
TUDelft

# Lecture Structure

1. Systems Engineering for Aerospace Design
2. Systems Engineering Methods
3. Verification and Validation I
4. Verification and Validation II
5. Risk Management
6. Concurrent Engineering
7. Requirements & Functional Analysis I
8. Requirements & Functional Analysis II
9. Handling Design Iterations
10. Exam information & Exercises



Following the lectures is crucial to succeed in the course!
Study of lecture slides is not sufficient!

# Lecture schedule

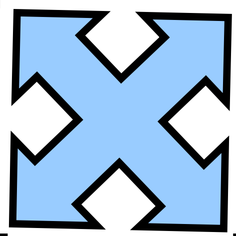
| Num. | Topic | Instructor | Date | Hour | Location |
|---|---|---|---|---|---|
| #1 | SE for Aerospace | E. Gill | 16/Feb | 8.45 - 10.45 | LR-CZ A + CZ C |
| #2 | SE methods | E. Gill | 17/Feb | 8.45 - 10.45 | LR-CZ A + CZ C |
| #3 | Risk Management & Concurrent Engineering (Design for Lifecycle) | E. Gill | 23/Feb | 8.45 - 10.45 | LR-CZ A + CZ C |
| #4 | V&V for S/C Control | E. Gill | 2/Mar | 8.45 - 10.45 | LR-CZ A + CZ C |
| #5 | V&V for S/C Propulsion | A. Cervone | 3/Mar | 8.45 - 10.45 | LR-CZ A + CZ C |
| #6 | W&B in aircraft | F. Oliviero | 7/Mar | 8.45 - 10.45 | LR-CZ A + CZ C |
| #7 | Requirements & Design for A/C Stability | F. Oliviero | 10/Mar | 8.45 - 10.45 | LR-CZ A + CZ C |
| #8 | Requirements & Design for A/C Controllability | F. Oliviero | 14/Mar | 8.45 - 10.45 | LR-CZ A + CZ C |
| #9 | Requirements & Design for A/C lateral and ground stability | F. Oliviero | 17/Mar | 8.45 - 10.45 | LR-CZ A + CZ C |
| #10 (TBC) | Recap (TBC) | F. Oliviero | 21/Mar | 8.45 - 10.45 | LR-CZ A + CZ C |

- Attendance to lecture is highly recommended even though recording will be made available
- Interactive exercise sessions will be done during some of the lectures with the purpose of practising the content (not replicable with the video recording…)

# Course Format

<u>Lectures</u> (20 lecture hours)
- Systems Engineering applied to aircraft and space missions
- *Active learning*

<u>Homework</u> (10 hours)
- Individual work on study material
- *Reflective learning*



<u>Tutorial</u> (~ 24 hours)
- Execution (2*12)
- *Collaborative learning*

<u>Exam</u> (~ 25 hours)
- Preparation (20)
- Execution (3)
- Reflection (2)
- *Reflective learning*

  TUDelft

# BrightSpace & Communications

## BrightSpace

- You need to enroll on BrightSpace for the course. Use self-enrollment!
- You have to use your TU Delft e-mail.
- Lecture slides are available on BrightSpace. There is no reader. Recommended books are indicated on the slides.
- Material for self study and tutorial will be placed on BrightSpace.

## Communications

- Communications to you is done via BrightSpace.
- All your questions shall be send to tudae3211.i@gmail.com.
- We will **NOT** respond to mails sent to other addresses.
- Send separate emails for space and aircraft; explicitly specify whether they are for *aircraft* or *space* or *organization* in the email subject.

# Assessment

| Assessment Element | Tutorial (Mandatory) | Exam (Mandatory) |
|---|---|---|
| Type | 2 X (Group homework) | Written exam |
| Weighting | 1/3 | 2/3 |
| Setting | Group | Individual |
| Deliverable 1 | 2 X Tutorial Report (per group) | Written exam |

Document with detailed information and instruction will be made available on BrightSpace.
Please check constantly the course information on BS.

Tutorial
- Each student must do 2 tutorials: **1 on *aircraft* and 1 on *spacecraft*.**
- The tutorial is **NOT** preparing for the exam, but it assesses different aspects.

Exam
- Registration information will be released later on Brighspace once the format of the exam will be confirmed by the BoE.

Passing of tutorial AND exam is required to pass the course!

# EXAM

The exam will be done in a form of a **digital** exam on campus.

- It will consist of a single session of 180’ minutes.
- It will be a closed book exam.
- it will consist of a mix of multiple choice questions and open questions.

The exam intends to test the students on several Learning Objectives of the course. It will be divided into System Engineering, Spacecraft and Aircraft topics. Please note that when testing for System Engineering related topics, some knowledge learnt in previous courses (that represent the pre-requisite to attend the AE3211) might be involved.

Limited exercise material will be provided on BrightSpace. Some exercises will be discussed during the lectures.

Passing of tutorial AND exam is required to pass the course!

# TUTORIAL

The tutorial is a homework group activity used to assess Learning Objectives in a complementary way with respect to the exam.

- There will be 2 tutorial activities, 1 on a space mission topic and 1 on an aircraft design topic: students must get a pass on both.
- 2 Reports must be prepared; assessment will be performed on those two documents. All the deliverables to be included will be discussed in the “tutorial requirements” documents that will be released at the beginning of the tutorial session.
- Group consists of maximum 6 students and Group Self-enrolment will be open on BrightSpace at the end of the lectures series.

Please refer to assessment information document that will be made available.

# Assessment-Related Times & Deadlines

Tutorial

Tutorial  group self-enrollment available on Brightspace: **14th March**

Self-enrollment closure: **20th March** (no later than 17:00 CEST)

Tutorials assignment available on BrightSpace: **22nd March**

Tutorial Preparation Reports delivery deadline: **6th April** (no later than 17:00 CEST)

Exam

Exam Date : **12th April**

Exam Retake Date (TBC): **19th July**

Tutorial Retake (tbc)

Tutorial  group self-enrollment available on Brightspace: **26th  June**

Self-enrollment closure: **5th  July** (no later than 17:00 CEST)

Tutorials assignment available on BrightSpace:  **10th  July**

Tutorial Preparation Reports delivery deadline: **25th  July** (no later than 17:00 CEST)

# Systems Engineering for Aerospace


SpaceShipTwo

# Today you'll learn to ...

1. understand the course setting and its links and relevance within the Bachelor curriculum

2. describe the characteristics of **complex aerospace projects**
3. describe **stakeholders needs**, requirements and constraints
4. describe the **process and framework** of Systems Engineering.

# Contents

- Complexity
- System Theory
- Systems Engineering
  - Definition
  - Need
- Stakeholders
  - Types
  - Constraints
- Process and Framework
  - Key parameters
  - Risk
- Conclusions

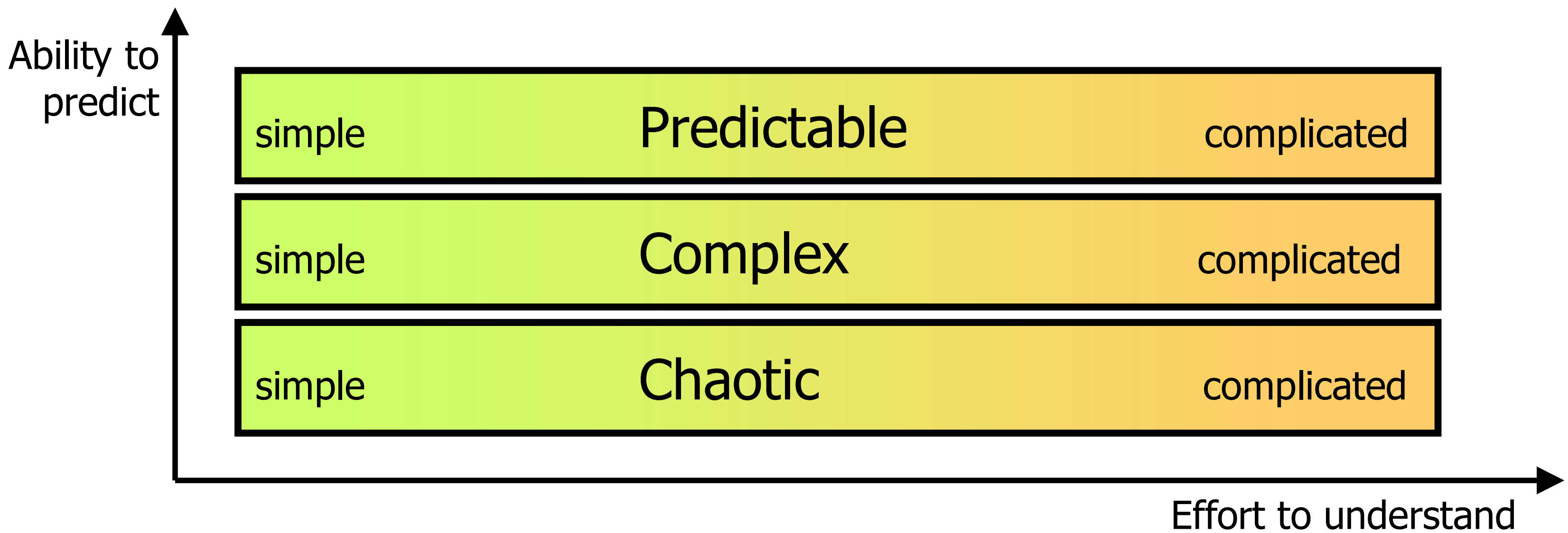
 

# Complexity

<u>Complexity</u> means something which comprises parts in **intricate** arrangement.

That arrangement makes complex systems more **difficult to predict** and introduces **uncertainty**.

Do not confuse <u>complex</u> with <u>complicated</u>! The latter means something which is not simple but still fully knowable.




Complex is the opposite of independent.

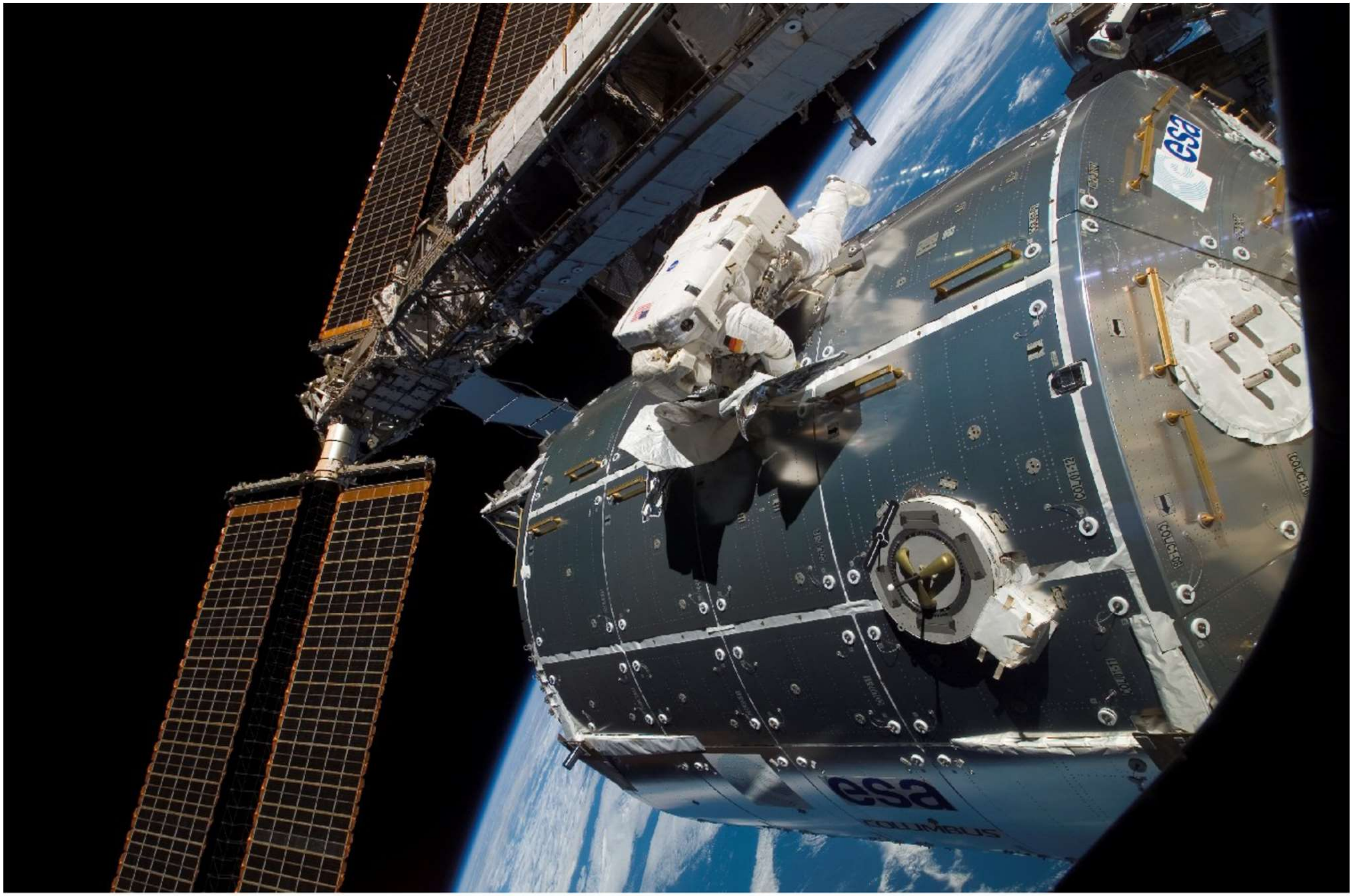
 

TUDelft

# Complicated Airbus A380 Harness

# Complex International Space Station




S122E008222

Source - EADS

# Exercise *Complicated versus Complex*

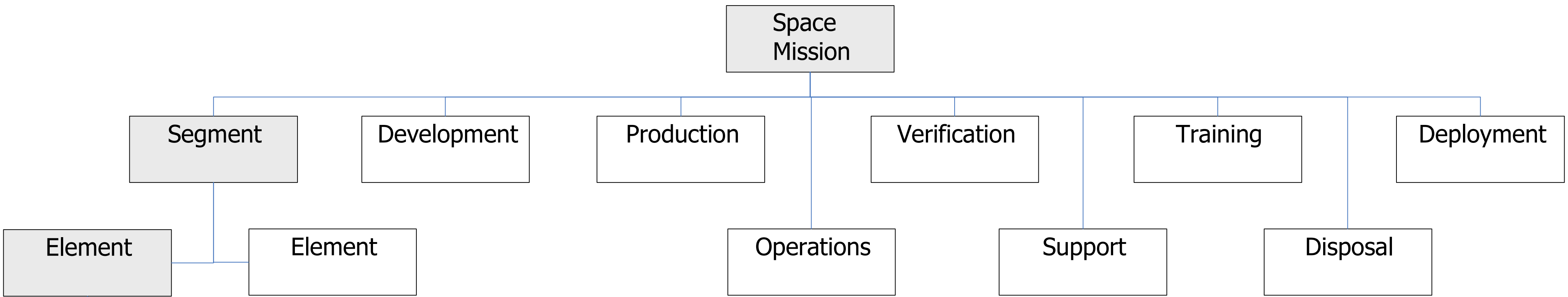
- The harness of an aircraft is …
- Designing the harness of an aircraft is …

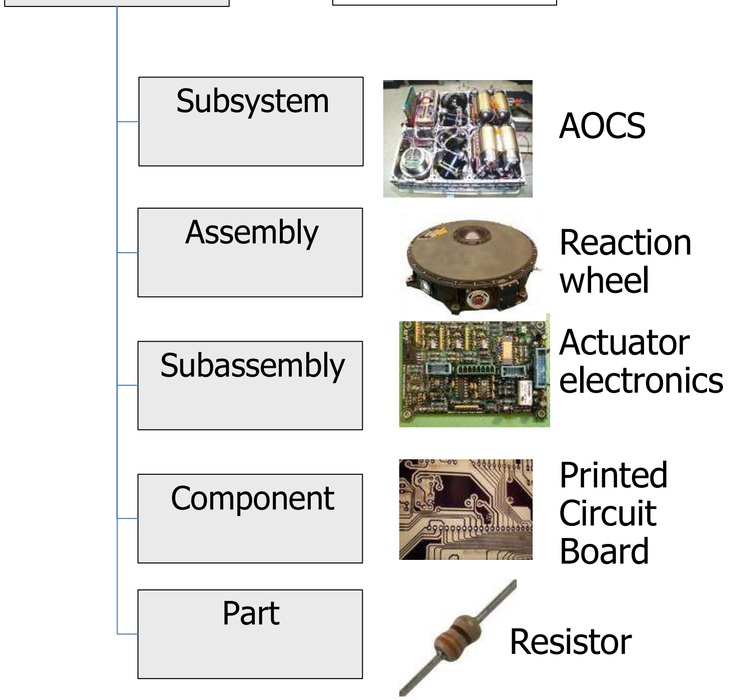
- My computer is …
- My software project is …

- My household is …
- My house is …

# What is a System?



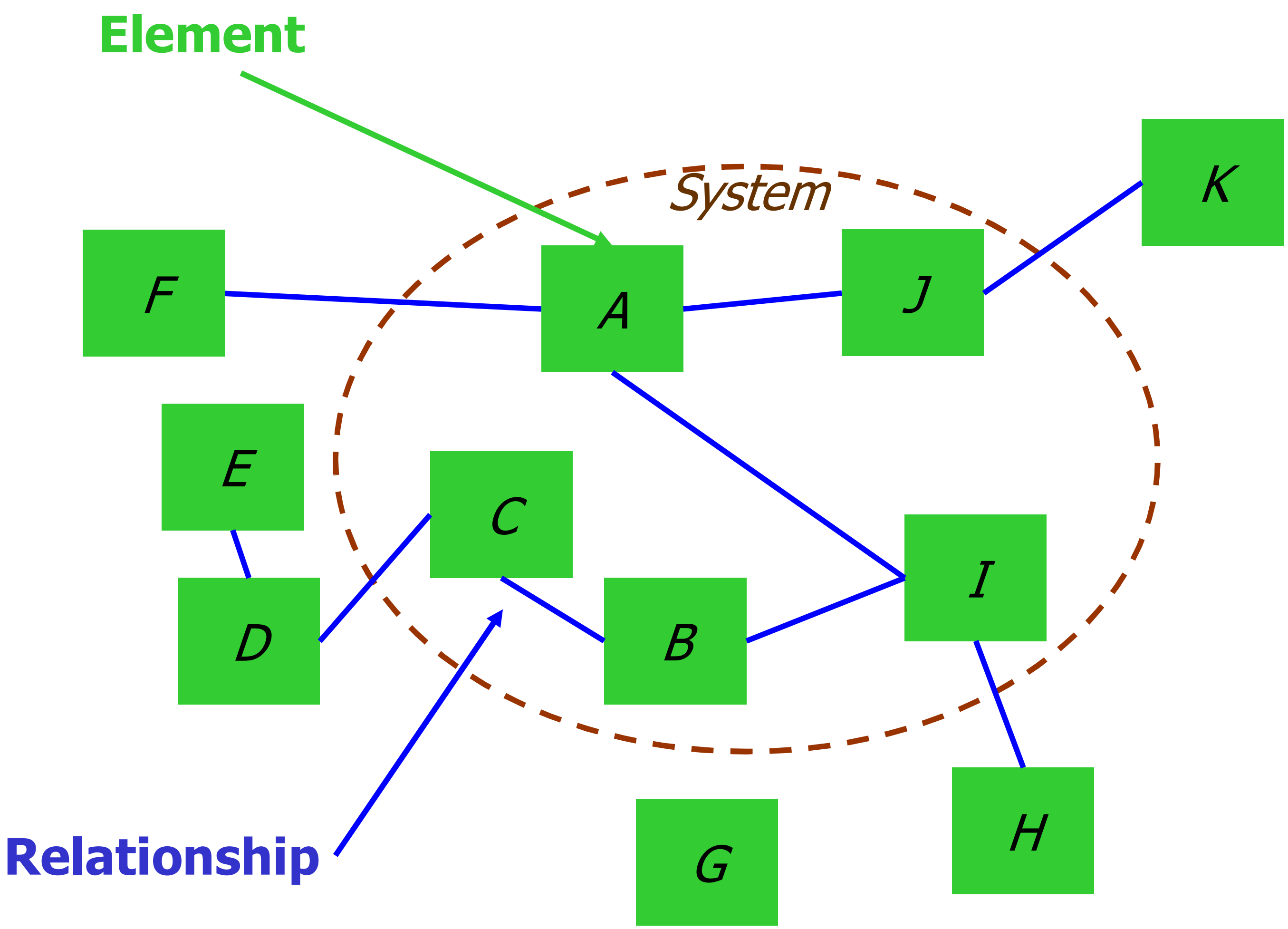




A system is a set of interrelated components **working together** to accomplish a common **purpose**.

AOCS: Attitude and Orbit Control System

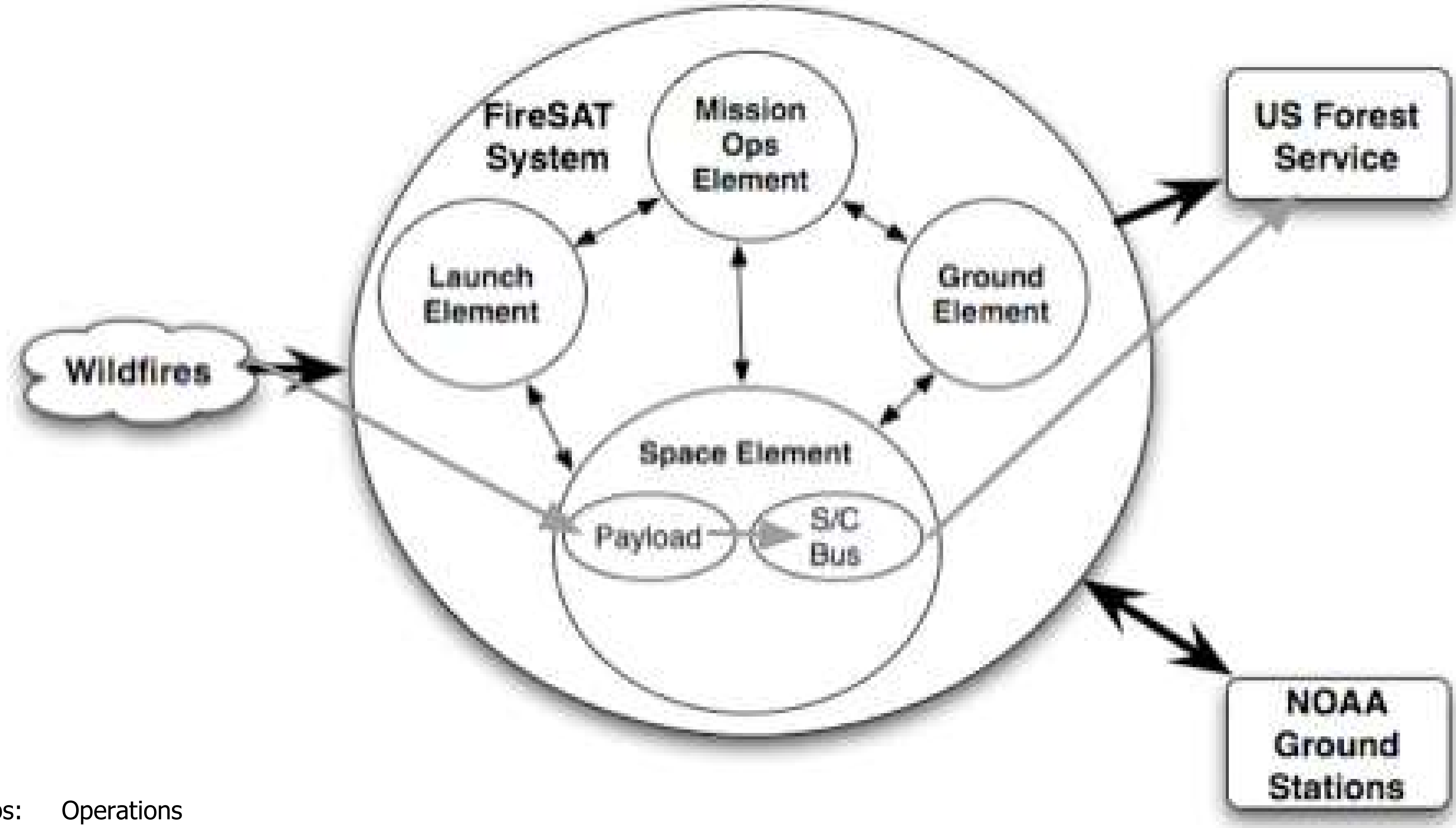
# System Description




A system can be described by its **elements** and **relationships** between its elements. Make sure that you clearly *define*, what the boundaries of your system are! What is outside of your system and has relationships to system elements is called **environment**.

# System Description Example: *FireSat*



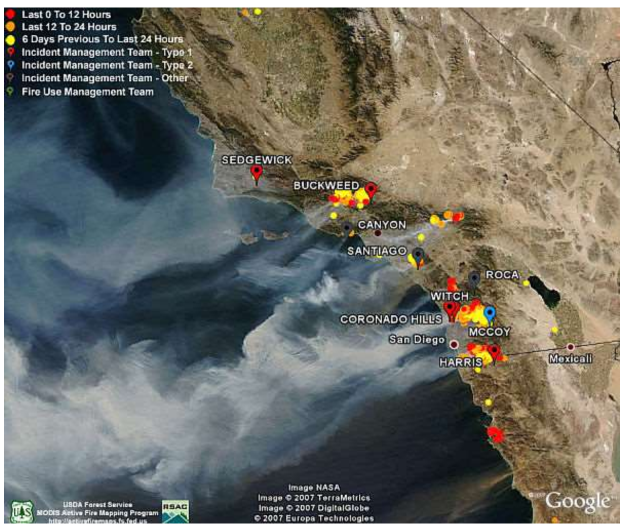

Ops: Operations
US: United States
S/C: Spacecraft
NOAA: National Oceanic and Atmospheric Administration

# What does the Space Element do?

A space element can be represented by a simple block.

You can zoom in to see more details, or subsystems in the block.

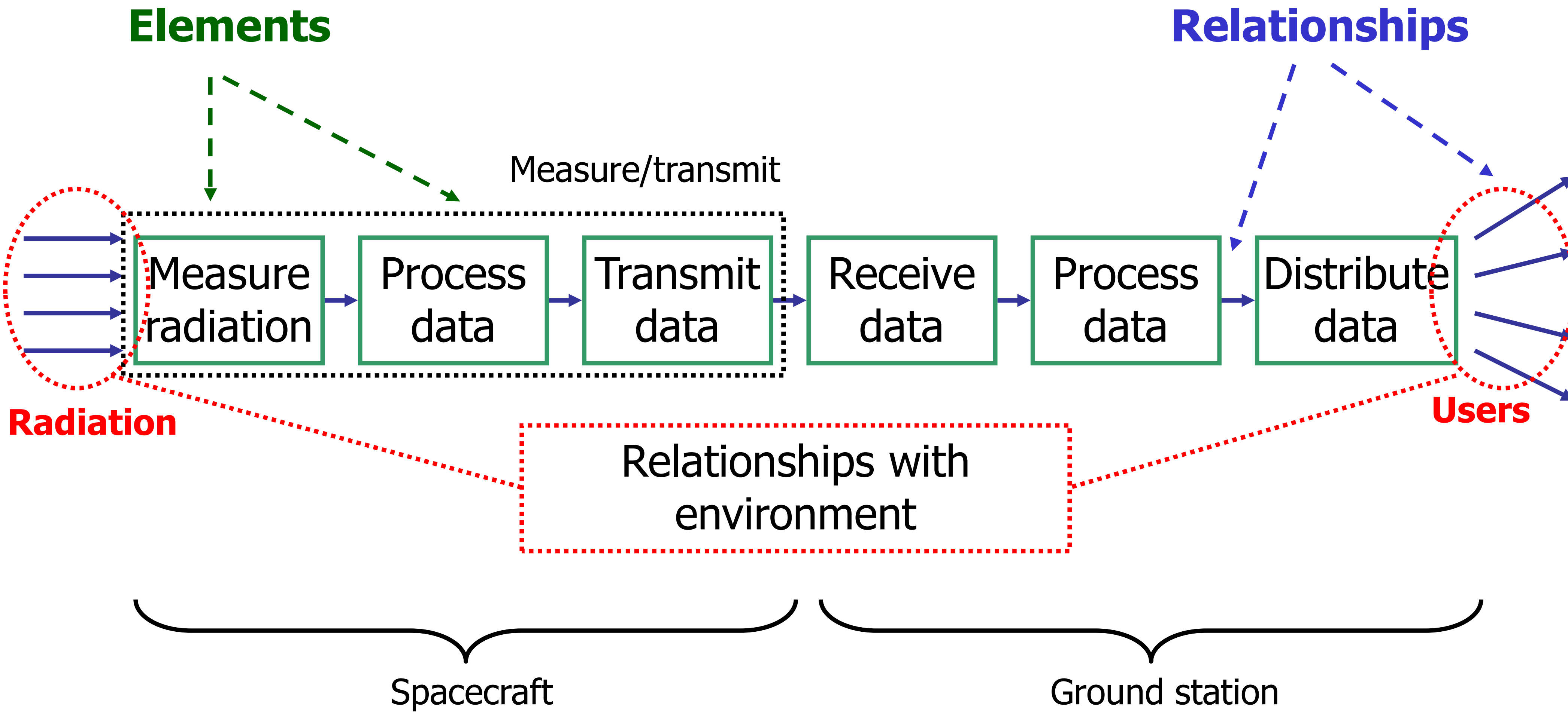



$\nearrow$ Measure/ transmit $\searrow$

A system description models **behavior**. It helps to support analysis and design.

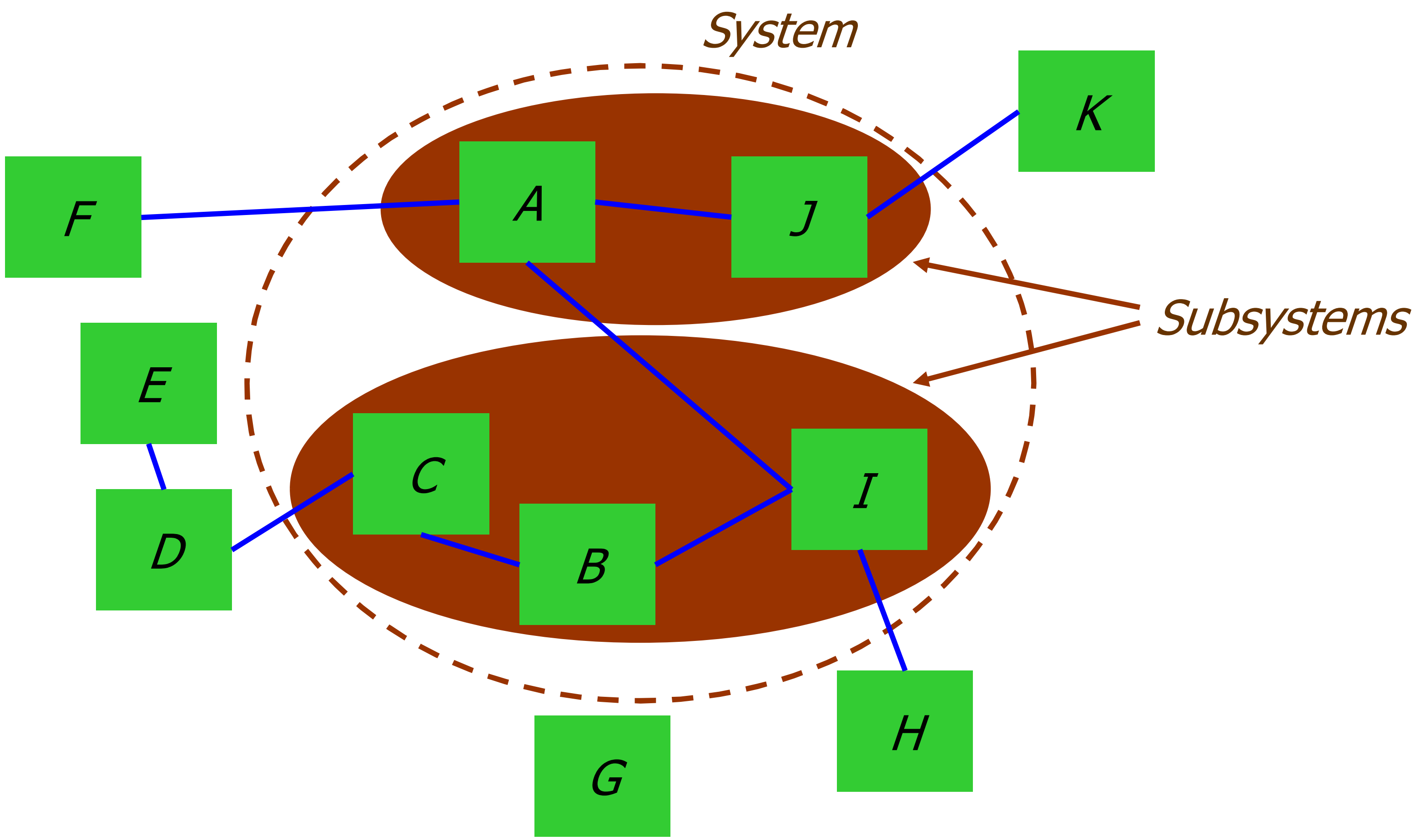
  TUDelft

# Zooming into the Functional Flow




The relationships with the environment determine the requirements on your system.

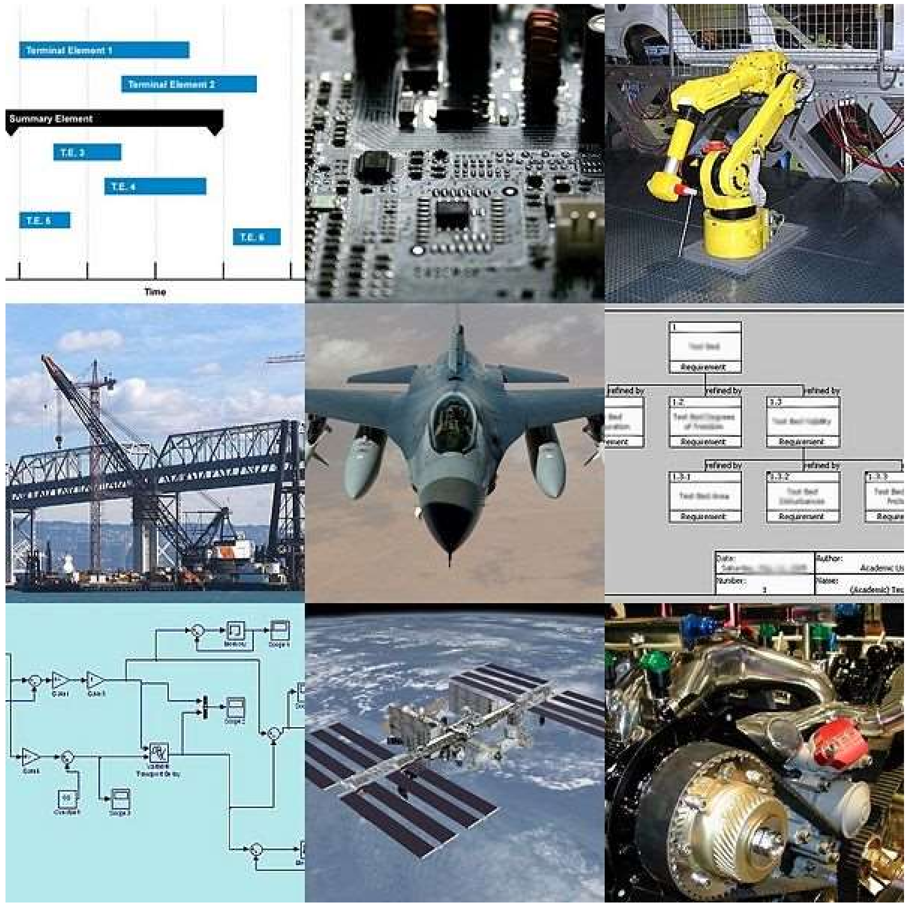
# Subsystems




A subsystem is a subset of elements in the system which retains the relationships between all elements. The same rules apply for subsystems as with systems.

# What is Systems Engineering?



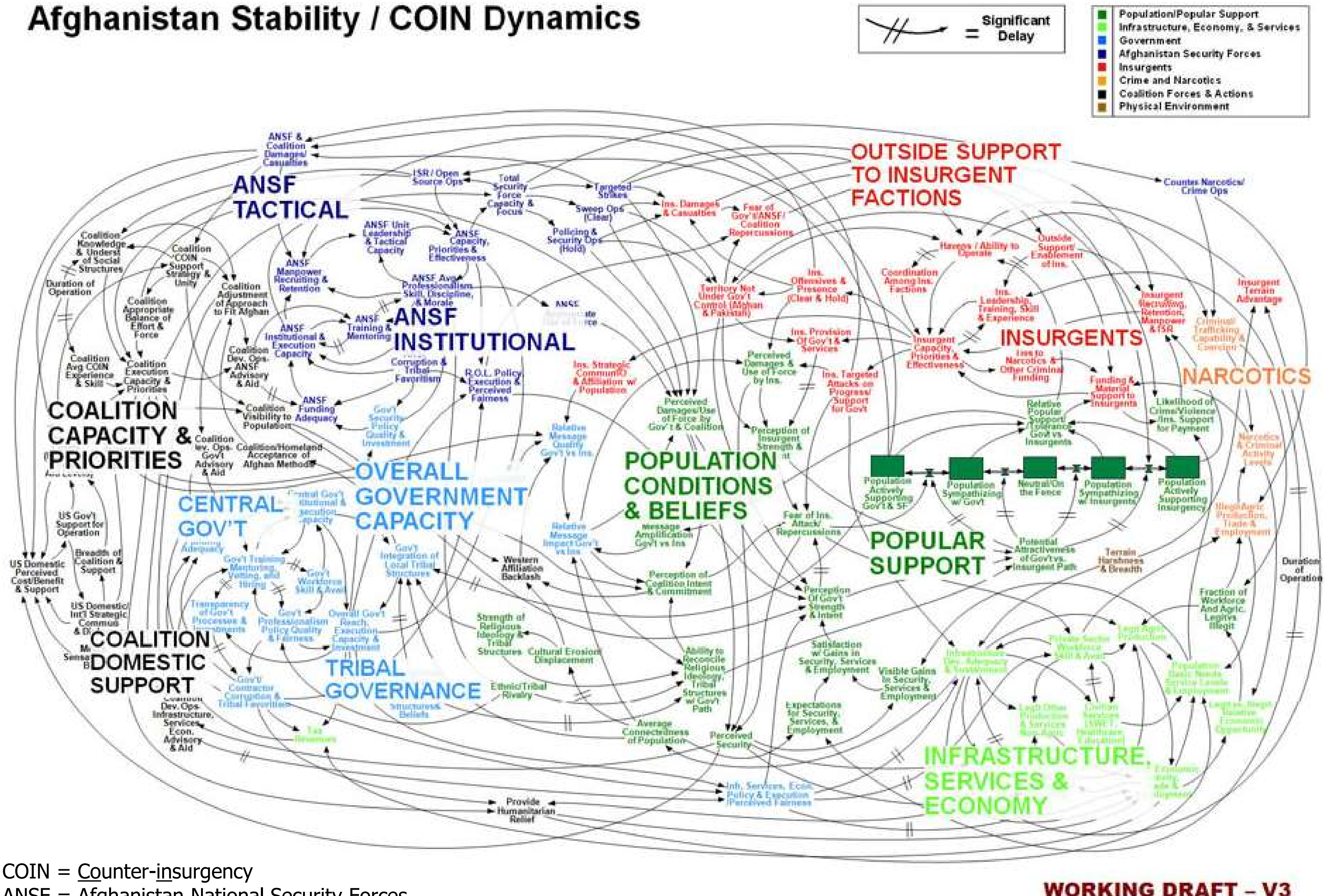

Source: Wikipedia

Systems Engineering is an **interdisciplinary** approach and means to enable the realization of **successful** systems.

Source: Incose

# Surprising application areas of Systems Engineering

## Afghanistan Stability / COIN Dynamics




COIN = Counter-insurgency

ANSF = Afghanistan National Security Forces

# Why Systems Engineering?

https://us-east-1.tchyn.io/snopes-production/uploads/2018/02/tesla-in-space.jpg
https://upload.wikimedia.org/wikipedia/commons/5/52

Rocket engineering is not like ditch digging. With ditch digging you can get 100 people and dig a ditch, and you will dig it a hundred times as faster if you get 100 people versus one. With rockets, you have to solve the problem of a particular level of difficulty; one person who can solve the problem is worth an infinite number of people who can't.

Source - Interview With Elon Musk Tesla Motors (2000s) https://en.wikiquote.org/wiki/Systems_engineering

  TUDelft

# Difficulty of Developing Successful Projects

In general industry, **for every 11 serious ideas** or concepts, 3 enter development, 1.3 are launched, and **1 succeeds**!

Reasons:

1. The better mousetrap that nobody wanted (technology driven; 28%)
2. The "me as well" product (too late; 22%)
3. Competitive one-upmanship (ignoring competition; 22%)
4. Instable architecture (15%)
5. Price crunch (13%)

Deficiencies in Systems Engineering contribute to about 45% of failed products.

Source – D. Verma, SpaceTech Lecture Notes (2004).

# Where Projects come from: Customers and their Interests

There are two main (abstract) reasons, which cause new projects:

Needs

- Functional deficiency
  - e.g. *global wildfire detection*
- Operational deficiency
  - e.g. *adding communication bandwidth*
- Changes in legislation and regulations
  - e.g. *Automatic Identification System (AIS) requires monitoring of ship routes*

Opportunities

- Technology fusion or breakthrough
  - e.g. *use of ion engines for operations of geostationary satellites*
- Discoveries
  - e.g. *potential ocean under the crust of Jupiter's Europa moon*
- Market
  - e.g. *space tourism*

A customer is the one who purchases a commodity or service.
Who are the customers in the above examples?

# Crucial for any Project: The Stakeholders

Customers are just a smaller part of a larger set of **parties who affect or can be affected** by a system. These are called <u>stakeholders</u>.


Specific stakeholders (*examples from aeronautics*):

- Customer (*airline*)

- User (*passenger*)

- Other stakeholders (*airport, residents close to airport*)

| |
|---|
| Understanding the needs of stakeholders is of **utmost importance** to the **success** of **any project**! |

  TUDelft

# Why Dealing with Stakeholders?

Identify and characterize stakeholders to:

- Overview the involved players
- Define the system boundaries
- Understand the complex problem space

Develop and analyze **stakeholder requirements** to:

- Obtain a common understanding of what their need is
- Avoid to miss any needs and to use implicit assumptions
- Synthesize a common set of requirements and identify priorities and conflicts

A requirement ("shall statement") is a singular documented need of what a particular product or service shall do or be.

Example Stakeholder Requirement:

*FIREBIRD-SH-3.1 The service shall be able to detect wildfires.*

*Example System Requirement:*

*FIREBIRD-6.2 The satellite shall have a dry mass of 100 kg max.*

# Exercise *Stakeholders @ LR*

Identify stakeholders in the educational system of the faculty of Aerospace Engineering!

- …
- …
- …
- …
- …

Identify the most relevant 3-5 ones (**Key stakeholders**)?

1. …
2. …
3. …

Key stakeholder are identified to know what to focus on and not to loose overview. Stakeholder requirements must address, however, ALL stakeholders.

# Stakeholder Needs

- After we have identified the stakeholders, we express <u>their needs</u> in stakeholder need statements. These <u>are translated in stakeholder</u> (or mission) <u>requirements</u> (a more formal way of expressing need statements).

- Stakeholder requirements must originate from an **understanding** of their needs. This preferably requires **interaction** (e.g. interview) with them.

- This is more difficult than you may think, because stakeholders…
  - … may not know exactly what they want
  - … may not provide you with all necessary information
  - … may hide <u>real needs</u> behind articulated needs
  - … may use different jargon and terminology
  - ….

# Properly expressing Stakeholder Needs

- Stakeholder needs must be solution-free!
  - They must express the "What", not the "How".
  - Use words similar to "have" and "do", instead of "for".
  - Example:
    - *Wrong: ~~"The Forest Service has a need for a fire detection satellite system."~~*
    - *Correct: "The Forest Service shall effectively detect and monitor potentially dangerous wildfires."*

- Stakeholder needs must be formulated in their "language"!
  - If not done, the need might not be able to be validated.
  - Example:
  - *Wrong: ~~"The car must have a drag coefficient less than 0.2."~~*
  - *Correct: "The car must have a cool look."*

Stakeholder needs and requirements are **high-level requirements** expressed in the **"language" of the stakeholder**, NOT in the "language" of engineers.

# Exercise *Airline Passenger Needs*

Identify airline passenger needs!

- …
- …
- …
- …
- …

Identify the most relevant 3-5 ones (**Key stakeholders requirements***)?

- …
- …
- …

Stakeholder *needs* can be expressed using different verbs (e.g. need, must).
All (stakeholder) *requirements* * MUST be formulated using the verb "shall"!

# Constraints

- Once we know the stakeholder requirements, we have to identify constraints.
- A constraint is a **restriction on the degree of freedom** in providing a solution.
- A constraint can be formulated as a requirement.

Constraints originate from …

**Economics**
e.g. *Firm Fixed Price Contract*

**Law and regulations**
e.g. *Frequency licensing*

**Technical**
e.g. *Use of space-proven hardware*

**Environmental**
e.g. *radiation*

**Culture**
e.g. *Company philosophy*

… and may impact…

**Resources**
e.g. *Cost*

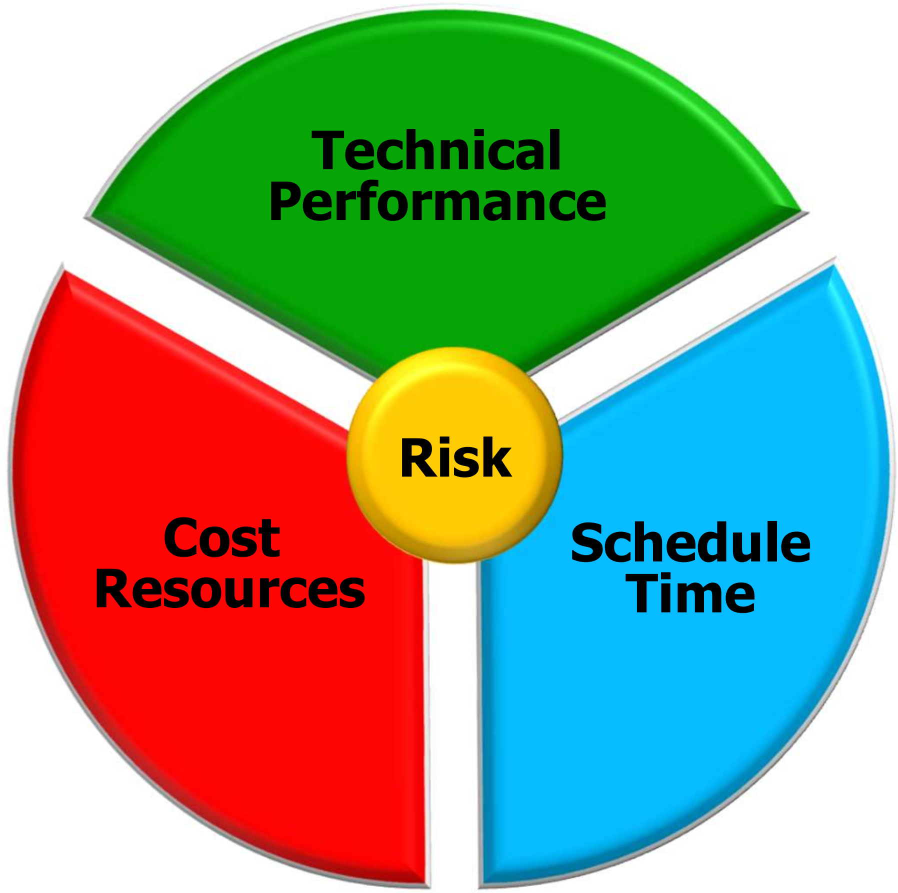
**Schedule**
e.g. *launch window*

**Functional performance**
e.g. *Resolution*

**Design process**
e.g. *model and testing approach*

**Project setting**
e.g. *Partners and Subcontracts*

# The Systems Engineering Universe




The Systems Engineering universe is established by technical performance, cost and schedule. All three dimensions are interconnected via **risk**.

An entire double hour on risk is provided in Lecture #8.

# Exercise *Check Proposition on Balancing*

Proposition: Good design has to continuously **balance** technical performance, cost and schedule.


Hint: Check this proposition by considering what will happen, when you reduce the risk in one area, with the risk in the two other areas. Do that for each of the three areas!

# The Systems Engineering Process

- We have identified the reason for applying Systems Engineering ("*why*") and the need of the stakeholders ("*what*").

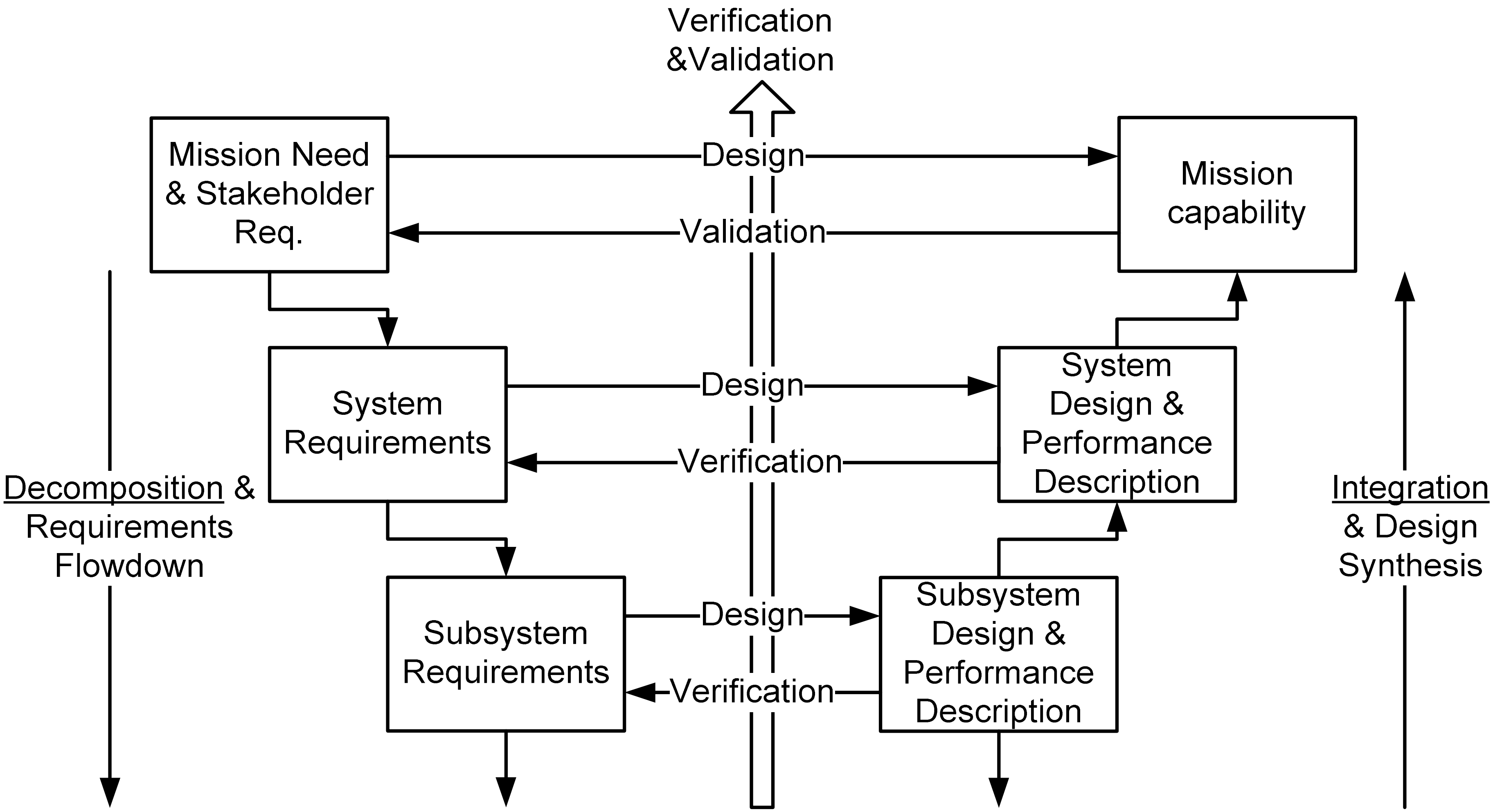
- To develop successful products, we still need to understand the "*how*". This is depicted in a **model** of the Systems Engineering process.

The activity of Systems Engineering (SE) can be understood as using a variety of SE models, methods, and tools along a single process of SE.

An entire double hour on SE methods and process is provided in Lecture #2.

# V-Model of the Systems Engineering Process




Various models exist of the Systems Engineering process, each with a specific focus.

TUDelft

# Summary and Conclusion

We have …

- characterized **complex aerospace projects**
- described the scope of systems and **systems engineering**
- identified **stakeholders** and **their needs, requirements** and constraints
- introduced a framework and **model of the Systems Engineering process**

Exercises for Homework:

- Exercises with worked-out solutions for this lecture can be found on BrightSpace.

Appendix:

- General literature on Space Engineering

# Literature

- Wertz J.R., Larson W.J.; Space Mission Analysis and Design; Third Edition; Microcosm, Inc. (1999)

  The classics on Space Mission Analysis and Design (SMAD) with excellent content.

  You can also use the “New SMAD”.