

25<sup>th</sup> anniversary  
annual INCOSE  
international symposium  
Seattle, WA  
July 13 - 16, 2015



# System Interface Engineering

Systems Engineering • Training • Coaching • Consulting



**Dieter Scheithauer**  
Dr.-Ing., INCOSE ESEP

Breitensteinstr. 26  
83727 Schliersee  
Germany

Phone: +49 (0) 80 26 - 97 68 00  
Fax: +49 (0) 80 26 - 97 67 99  
Mobile: +49 (0) 170 - 23 50 23 4

[www.hitseng.eu](http://www.hitseng.eu) • [dieter.scheithauer@hitseng.eu](mailto:dieter.scheithauer@hitseng.eu)



# Content



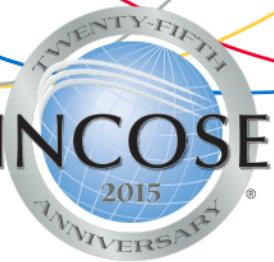
- Introduction
- System Interface Engineering Scenarios
- System Interface Engineering Terminology
- Interface Control Documents
- Product Taxonomy
- Conclusions

## Note:

*The paper itself describes the integration of system interface engineering into the system development flow in addition. Due to the dense level of detail, this part of the paper is omitted from the presentation.*



# The Flow of Configuration Baselines

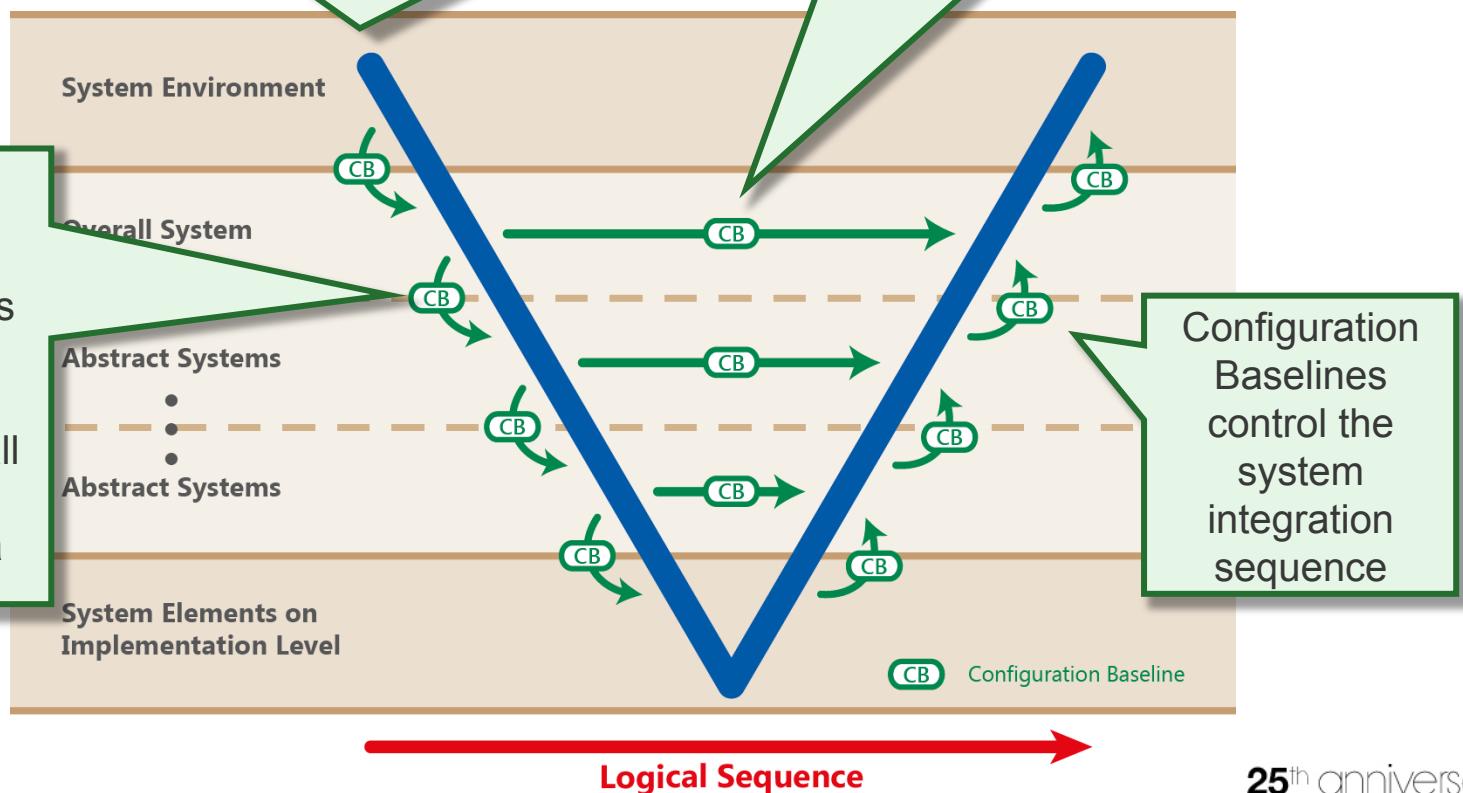


Configuration Baselines control and safe guard the flow of information between system and system elements across the system architecture

Configuration Baselines establish the basis for preparing system integration

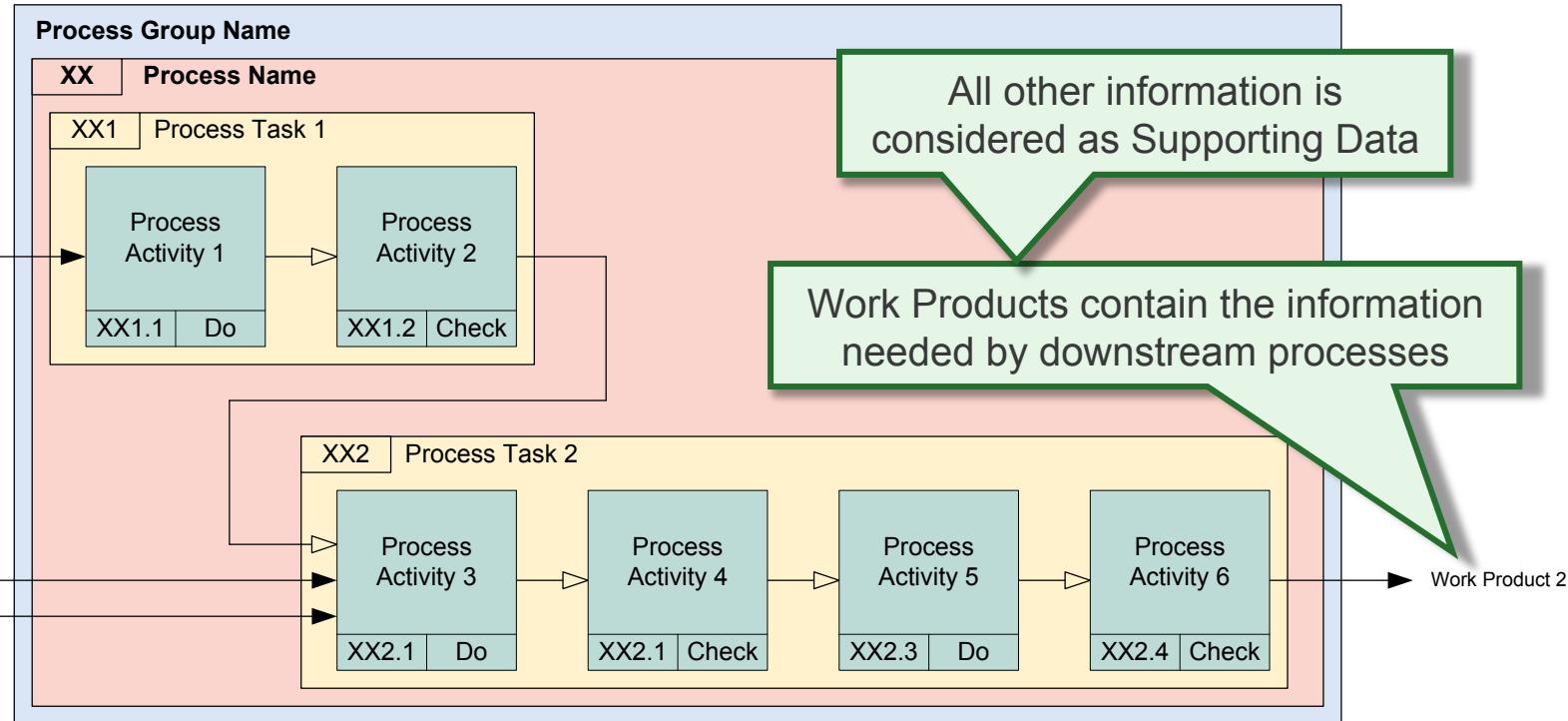
Configuration Baselines ensure consistent sets of allocated requirements forwarded to all the system elements of a system

Configuration Baselines control the system integration sequence





# Work Product Generation Sequence



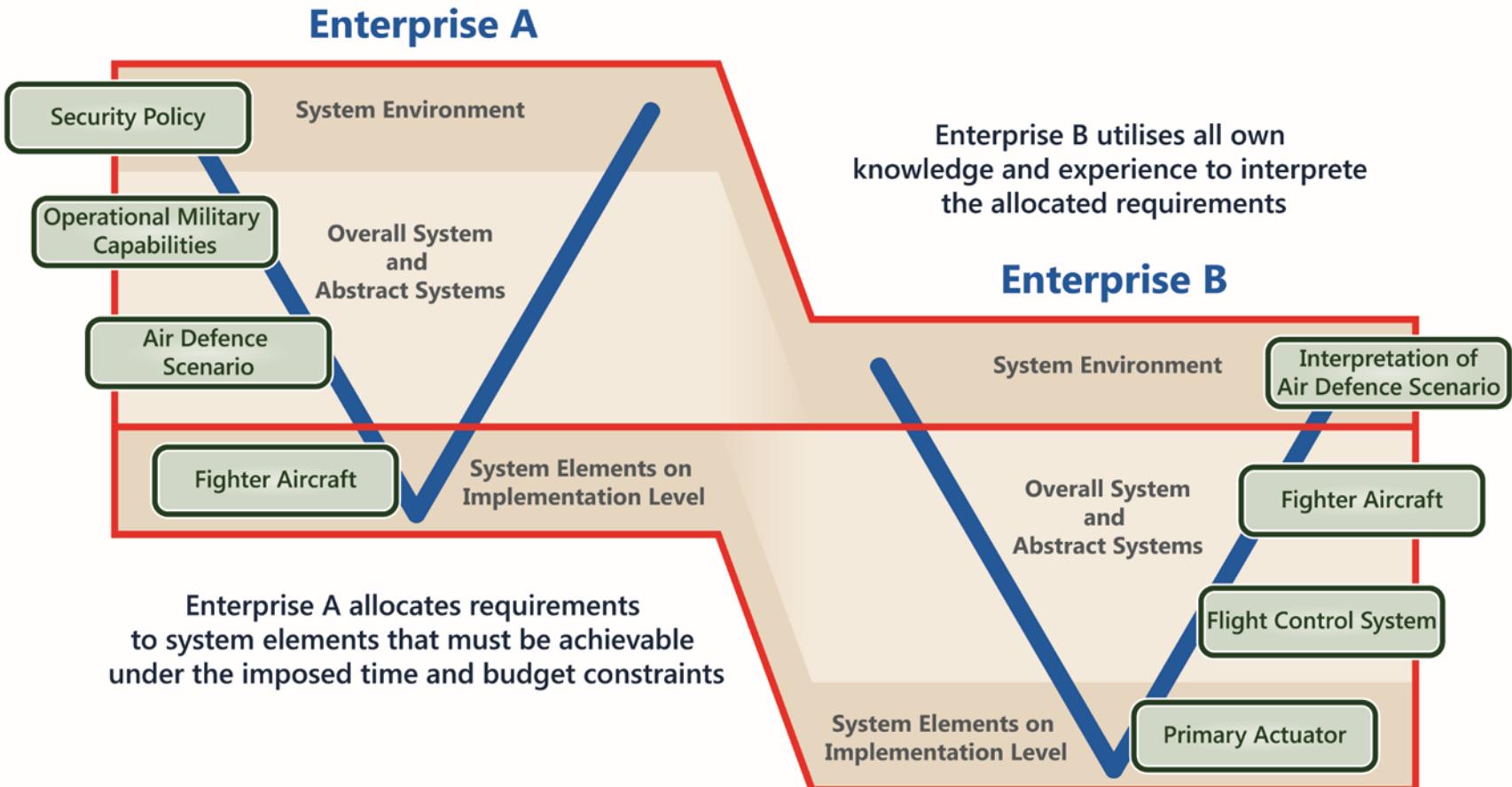
## Process Definition Model

- Four distinct architectural levels
- Each level featuring specific semantics
- Supporting a well balanced process definition

Work Product Generation Sequences describe the flow of Work Products for generating consistent Configuration Baselines

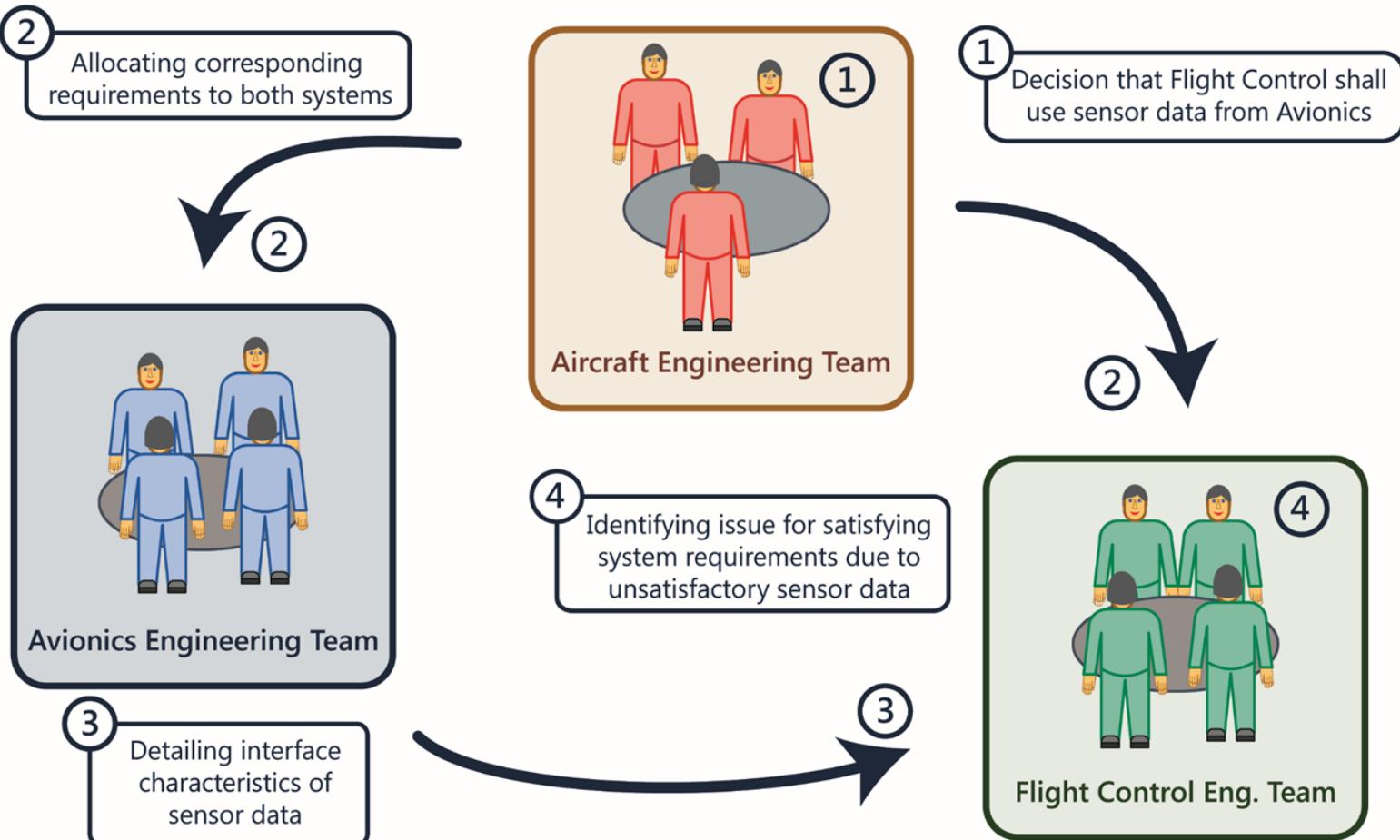


# Organisational Workshare





# Scenario A





# Was the Avionics Team Right?



- Every design team in charge of developing a system, has a responsibility for the quality of their system
- The Aircraft Engineering Team concentrates on the emergent properties and features on their respective level, and defines clear and feasible allocated requirements for their system elements
- The Aircraft Engineering Team has no obligation to define further details of the interface between avionics and flight control
- It is a shared responsibility of the Avionic Engineering Team and the Flight Control Engineering Team to specify the interface's details
- The involvement of the Aircraft Engineering Team would not add value as detailing the interface characteristics has no impact on their design decision
- Consequently, the Avionic Engineering Team has the right to propose details of the particular sensor data interface



# Alternative Ways of Conduct



- Synchronized Decision Making

- In an ideal world it would be preferable that the detailing of the interface characteristics is done together by the Avionic and Flight Control Engineering Teams at one point in time
- However, that makes the slowest progressing team dominating the pace of overall development progress

- Asynchronous Decision Making with a Three State Logic

- The team proposing an interface refinement gets a timely response from the other team for agreement and proceeds with their development
- The team proposing an interface refinement gets a timely response from the other team for non-agreement demanding another valid solution
- The team proposing an interface refinement gets no timely response and may decide to proceed, or to go on hold



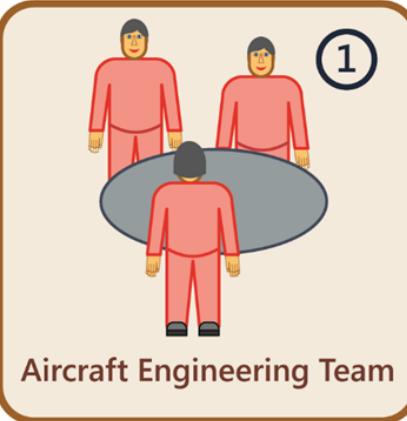
## Scenario B

There may be no engineering authority for the common higher level system of flight control and pilot training aids

2 Allocating requirements for agility and carefree manouvring to flight control

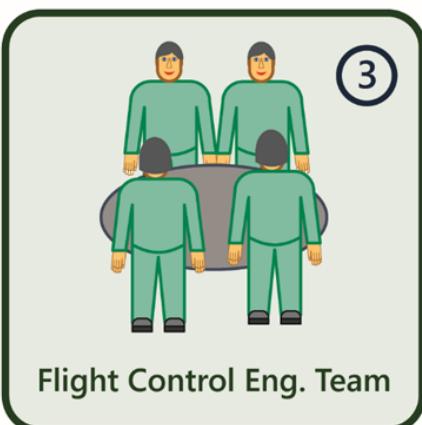


2



1

Decision for single pilot instead a crew of two for reducing aircraft mass and maximizing aircraft agility. In consequence, the pilot workload for basic flight control has to be minimized.



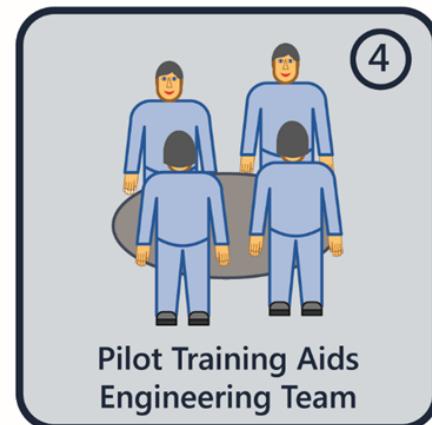
Flight Control Eng. Team



3 In balancing the competitive requirements regarding carefree manouvring and agility, the Flight Control Engineering Team proposes enhanced normal pilot's skills

4

The additional training demands may be in conflict with the flight simulator infrastructure developed by the Pilot Training Aids Engineering



The interface between the Flight Control Engineering Team and the Pilot Training Aids Engineering Team has no impact on lower level systems of the flight control system

anniversary  
al INCOSE  
international symposium  
Seattle, WA  
July 13 - 16, 2015



# Internal and External System Interfaces



From the viewpoint of a system, the interfaces to Neighbouring Systems and those defined by Higher Level Systems are external interfaces

Higher Level Systems

Upper Level System

System

System Elements

Lower Level Systems

Neighbouring Systems

Increased awareness of External Interfaces is an important factor in successful System Interface Engineering

From the viewpoint of a system, the interfaces between Lower Level Systems are Internal Interfaces

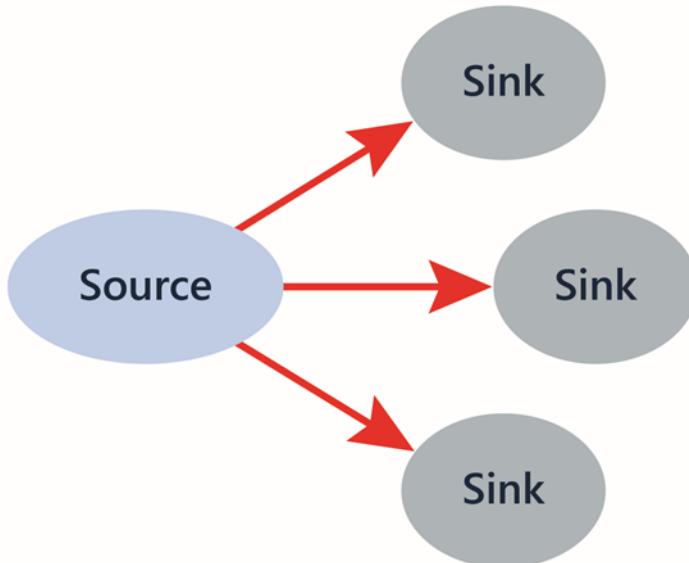
Neighbouring Systems



# Information Flows



- System interfaces are the results of the architectural decomposition throughout the whole system architecture
- For system interoperability considerations, interfaces are abstracted as information flows flowing from one source to one or multiple sinks

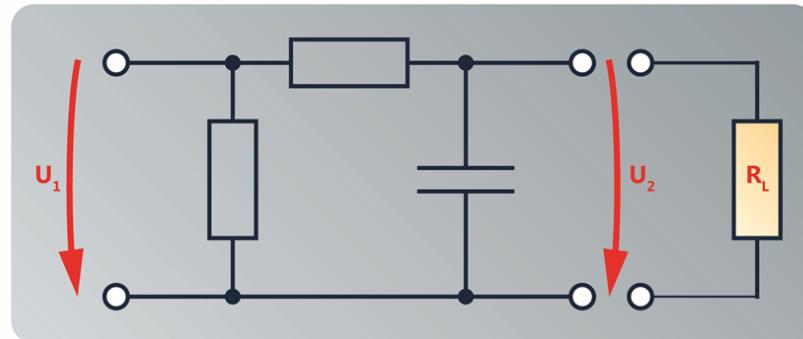
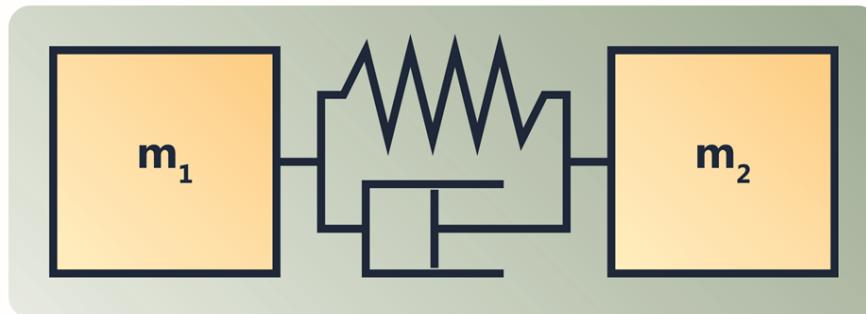




# Physical Interfaces



- Physical laws define mutual dependencies between input and output
- Unidirectional motion or flow of energy can be approximated only





# Human Communication



- Information flow is only one aspect of human communication
- System interface design needs to consider all aspects of human communication

25<sup>th</sup> anniversary  
annual INCOSE  
international symposium  
Seattle, WA  
July 13 - 16, 2015



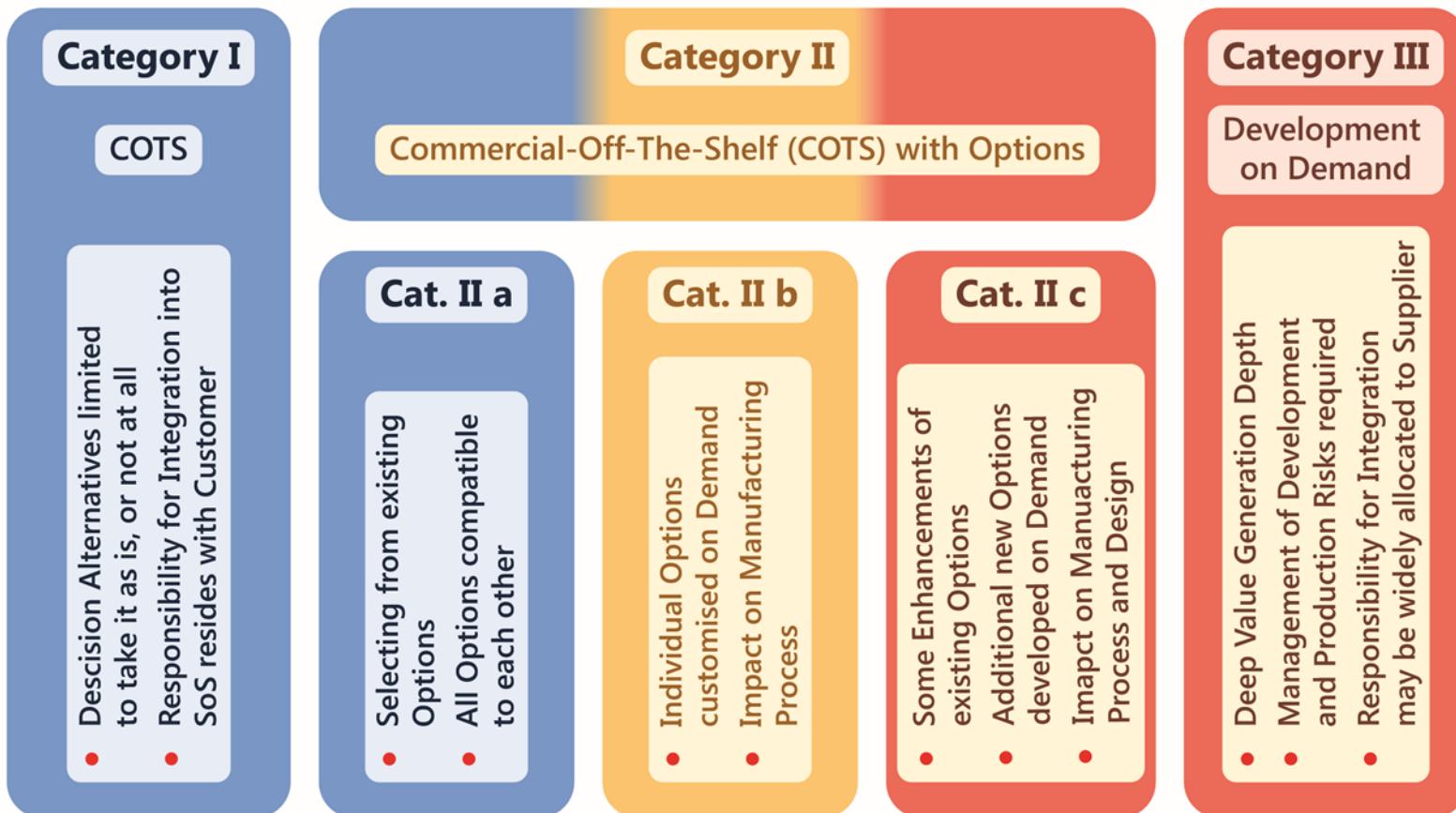
# Interface Control Documents



- All systems engineering text books mention Interface Control Documents (ICDs) when they consider system interfaces
- ICDs are of critical importance for contracting system elements on the implementation level
- The compilation of ICDs has a reputation as tedious tasks
- This reputation is mainly caused by the fact that the generation process of ICDs is at odds with any, even weak kind of value stream thinking
- It is proposed to manage the refinement of system interface requirements as integral part of the allocated and system requirements, and to generate ICDs as reports out of the requirement repositories



# Product Taxonomy



25<sup>th</sup> anniversary  
annual INCOSE  
international symposium  
Seattle, WA  
July 13 - 16, 2015



# Conclusions



- Neglecting system interface engineering, a perfect systems engineering process may give the impression of a pure deductive process
- Information models of interfaces are coarse approximations of the mutual dependencies present in natural laws, and the complexity of human communication
- For efficiency reasons, it is recommended to agree on interface refinements on a peer-to-peer basis
- This demands an improved awareness of each engineering team for their external interfaces to be considered by the systems engineering process definition
- ICDs may then be generated as reports out of the requirement repositories in order to check the completeness of the interface definitions before allocated requirements are handed over to system elements on the implementation level
- COTS enables interface standardization, but also constraints the level of freedom of engineering teams regarding design decisions



Thank You  
for your attention



Dieter Scheithauer  
Dr.-Ing., INCOSE ESEP

Breitensteinstr. 26  
83727 Schliersee  
Germany

Phone: +49 (0) 80 26 - 97 68 00  
Fax: +49 (0) 80 26 - 97 67 99  
Mobile: +49 (0) 170 - 23 50 23 4  
[dieter.scheithauer@hitseng.eu](mailto:dieter.scheithauer@hitseng.eu)  
[www.hitseng.eu](http://www.hitseng.eu)

**25<sup>th</sup> anniversary**  
annual INCOSE  
international symposium  
Seattle, WA  
July 13 - 16, 2015