25th anniversary
annual INCOSE
international symposium
Seattle, WA
July 13 - 16, 2015

Structuring Requirements in Standard Templates

Richard Beasley, Iain Cardow, Michael Hartley and Andrew Pickard

Rolls-Royce

Co-Authors

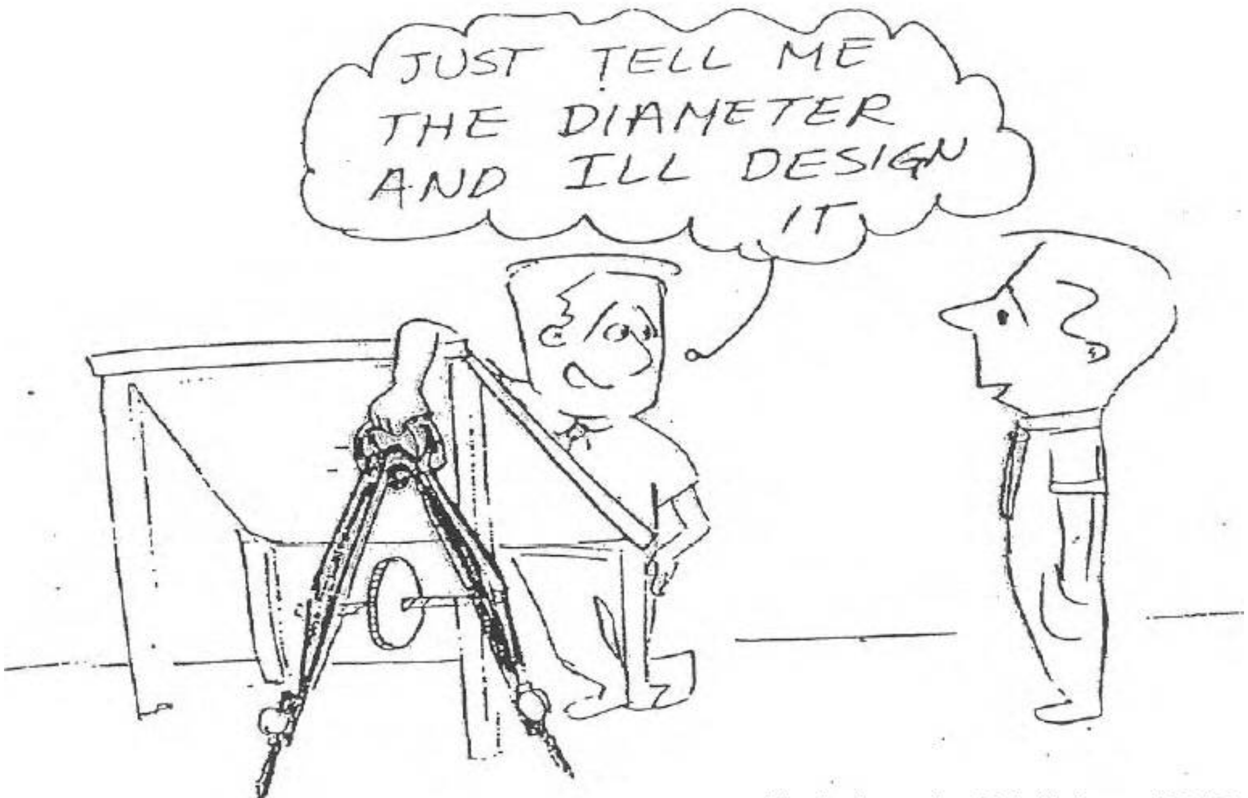
- Richard Beasley
- Richard.Beasley@rolls-royce.com
- Iain Cardow
- Iain.Cardow@rolls-royce.com

- Mike Hartley
- Mike.Hartley2@rolls-royce.com
- Andrew Pickard
- Andrew.C.Pickard@rolls-royce.com

Introduction

- This presentation summarizes the approach to standardized documentation and structure of requirements data taken in Rolls-Royce
- Presentation divided into sections as follows
  1. Why we needed to standardize, and what we needed to emphasize in this standardization
  2. Documentation structure, product structure and interfaces to support desired practice
  3. Standard requirements document structure and contents
  4. Standard definition document (derived requirements and product elements
  5. Conclusions

But First…



Nicely drawn by Rich Holmen, MDAC

- Need system design to be requirements led

1) Why standard? Why this standard?

- In Rolls-Royce we are trying to make Systems Engineering the way we do Engineering
- This includes a strong (but not exclusive) focus on requirements. Specifically:
  - **Generation of Requirements** – elicitation, completion and understanding to drive the design
  - **Requirements management** – actually requirements and derived requirements from definition and Verification data
- Emphasis of Systems Thinking to ensure complete and understanding of requirements – and focus on technical attributes and functions rather than separate stakeholders
- Design sub-elements in context to avoid sub-optimisation of the parts rather than optimization of the whole
- Good structure to support change in long product / service life of products
- Standardisation makes cross site working feasible

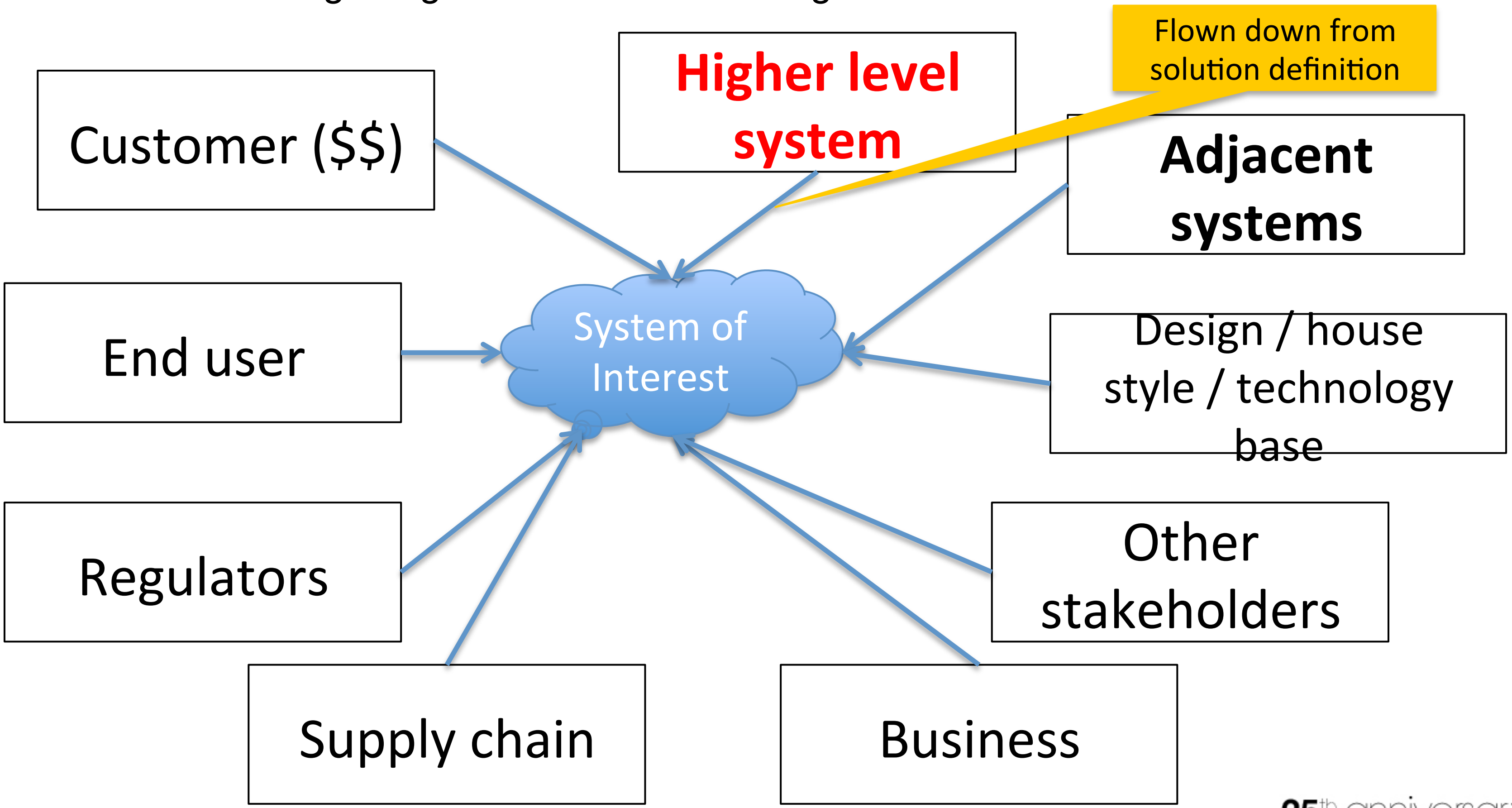
2) Desired Practice

1. For each layer / element of the system capture all stakeholder needs , analyse, and document as complete set of requirements
   – In standard **Requirements Document (RD)**
2. From solution, derive requirements for what other parts need to do for that element to work
   – In standard **Definition Document (DD)**
3. Manage requirements information in clear, **top down** structure – to enable change control by higher level integrating what is asked of its sub-elements

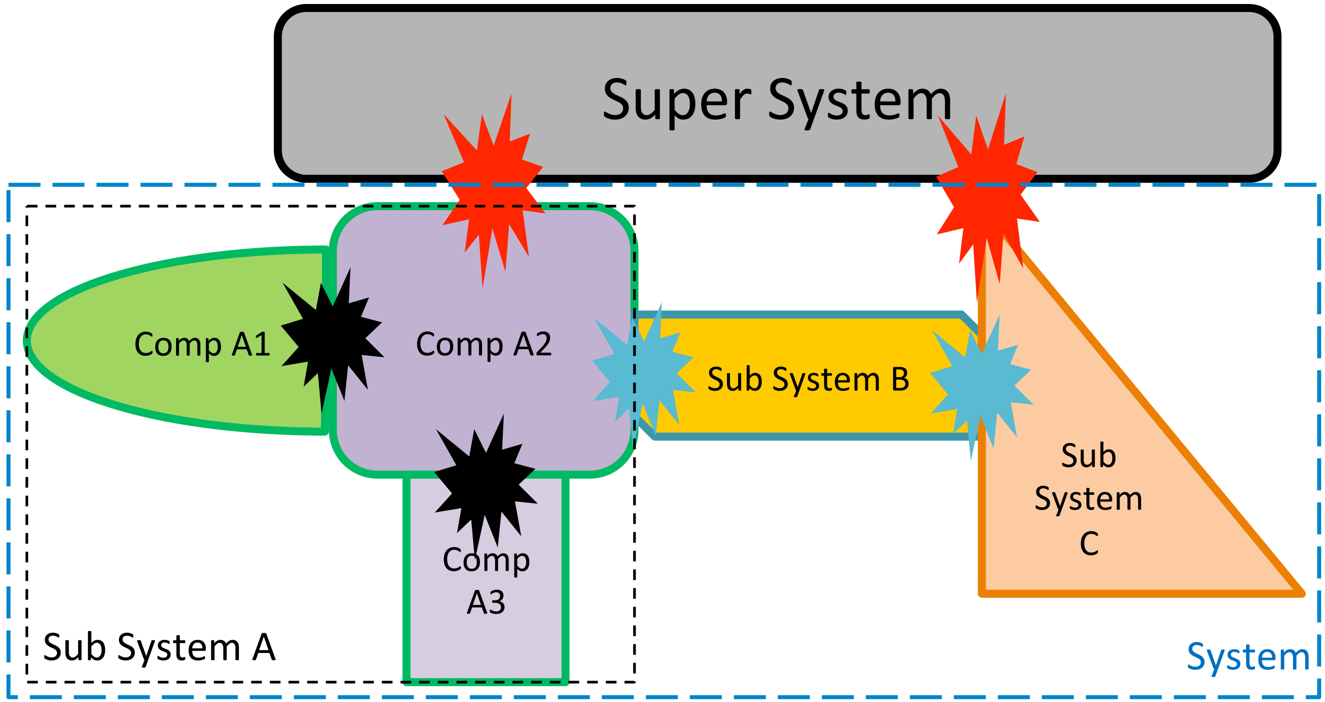
2a) Determine requirements

For every system element elicit all stakeholder needs, and apply Systems Thinking to analyse and complete –

- focus on integrating attributes and defining functions



Interface Flow and Reporting

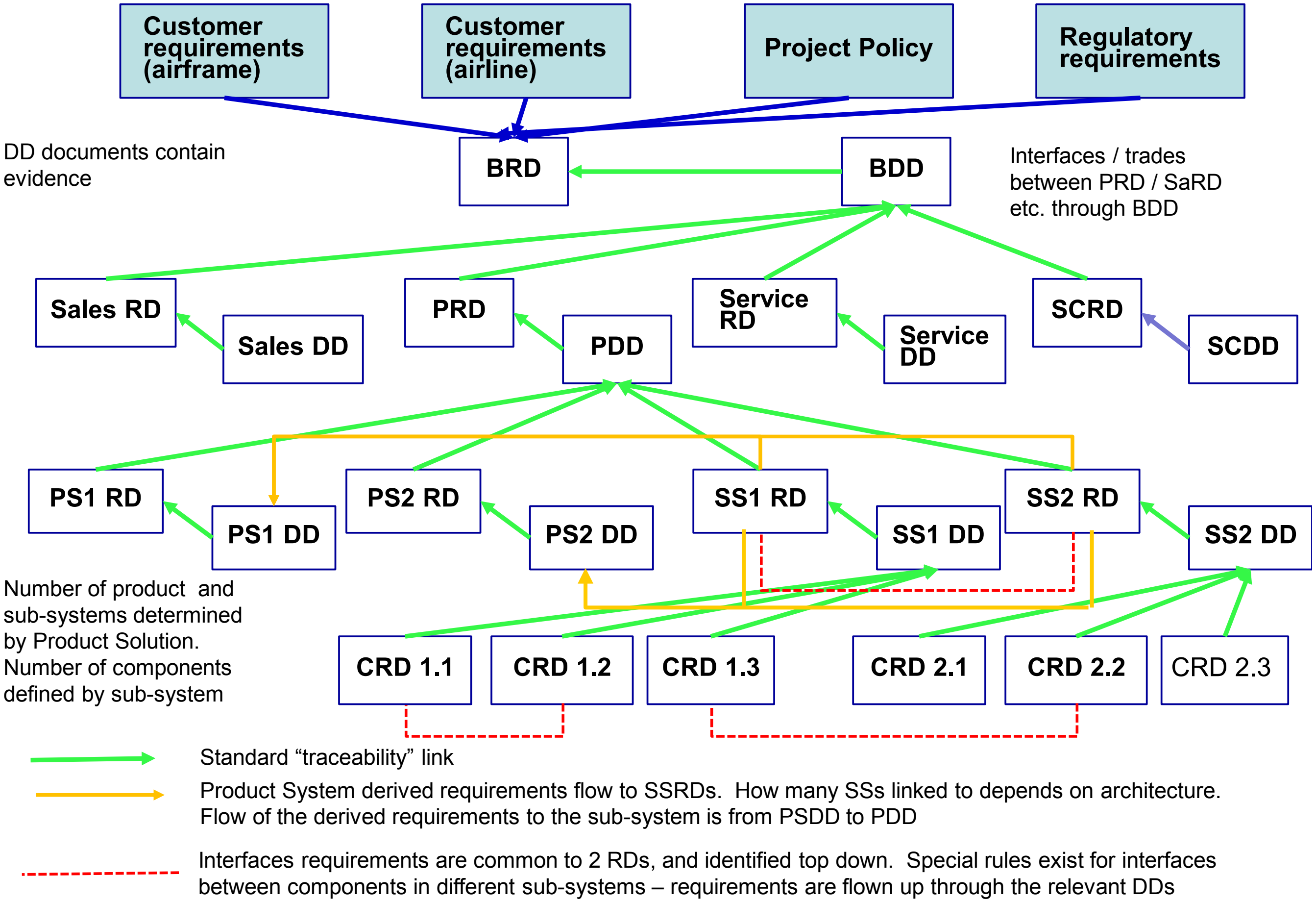


Super System DD Identified Interface Reflected in System RD

Sub-System DD Identified Interface Reflected in Component RD

System DD Identified Interface Reflected in Sub-System RD

Requirements and Definition Document Hierarchy for New Product Introduction



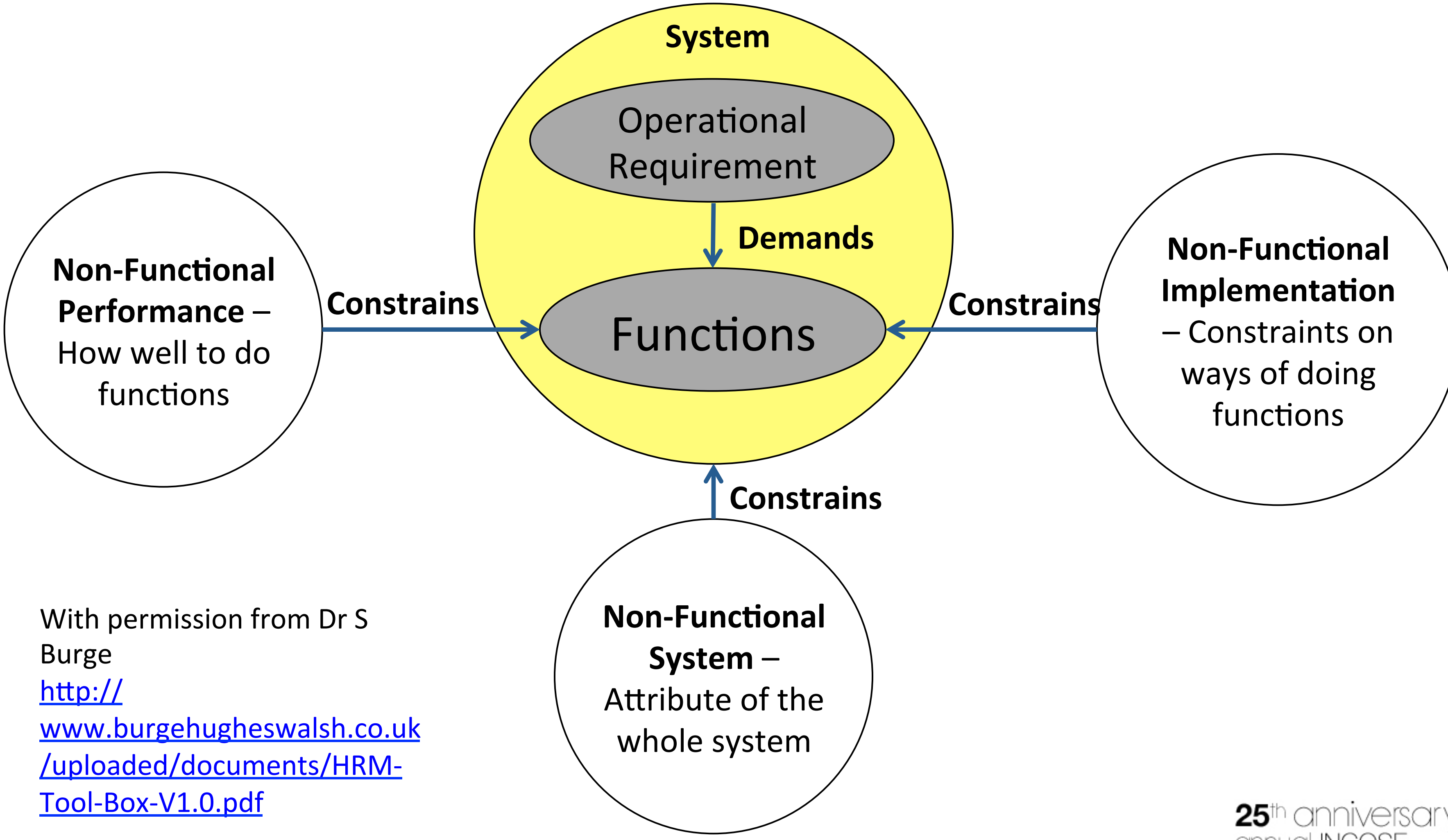
Issues affecting product from Sales, Services or Supply chain solution (or vice versa) are flown via the BDD to the relevant document

3) Requirement Documentation

- Requirements should be documented and captured
  - With uncertainty, assumptions and blanks allowed but clearly indicated
- Want to integrate different stakeholder needs around attributes and functions
  - We use "holistic requirements model" to distinguish requirement types
- Traceability back to the individual stakeholders is important

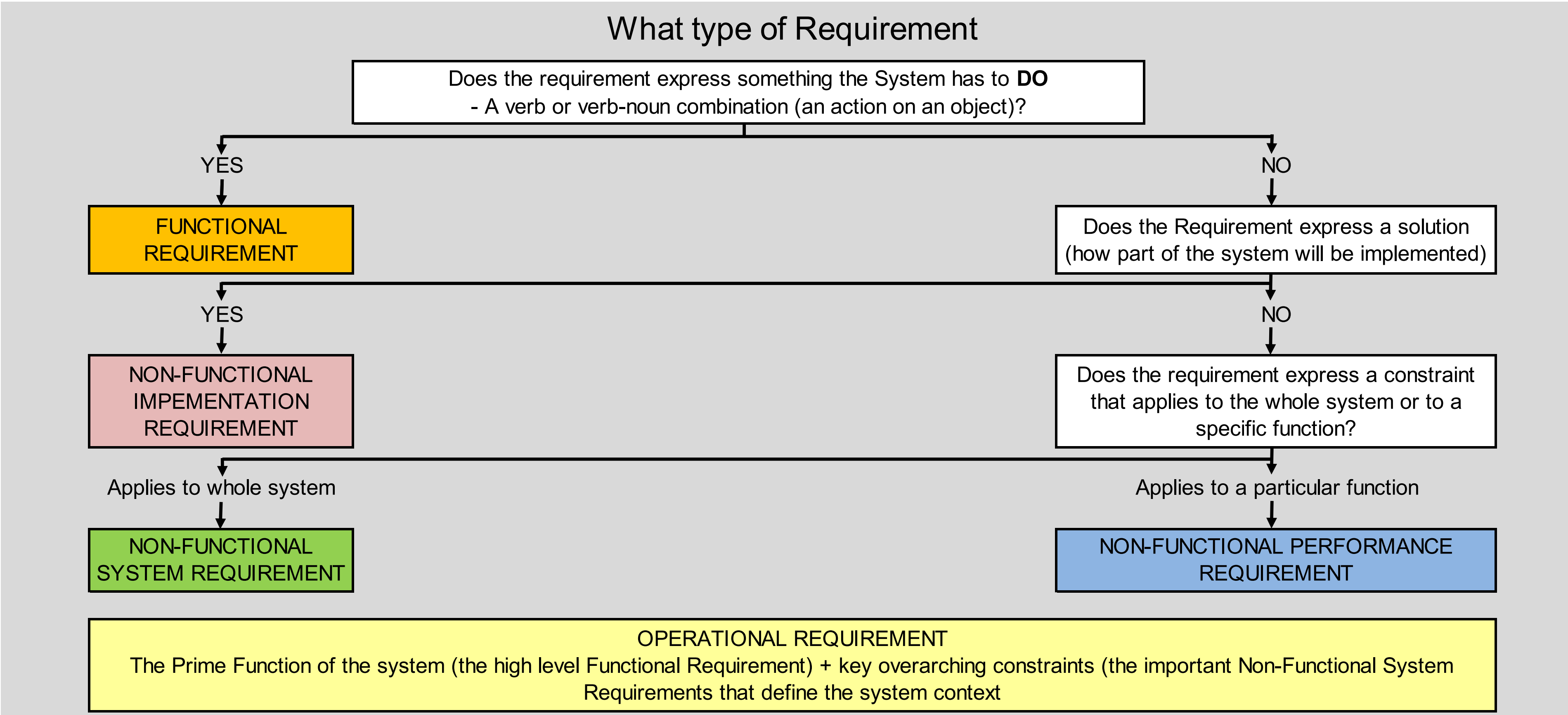
On-going documentation allows clarity of current understanding, included explicit recognition of what is unclear, uncertain or unknown

Holistic Requirements Model



With permission from Dr S Burge
http://www.burgehugheswalsh.co.uk/uploaded/documents/HRM-Tool-Box-V1.0.pdf

Determination of Type of Requirement



With permission from Dr S Burge

http://www.burgehugheswalsh.co.uk/uploaded/documents/HRM-Tool-Box-V1.0.pdf

Generic Requirements Document Content (Part 1)

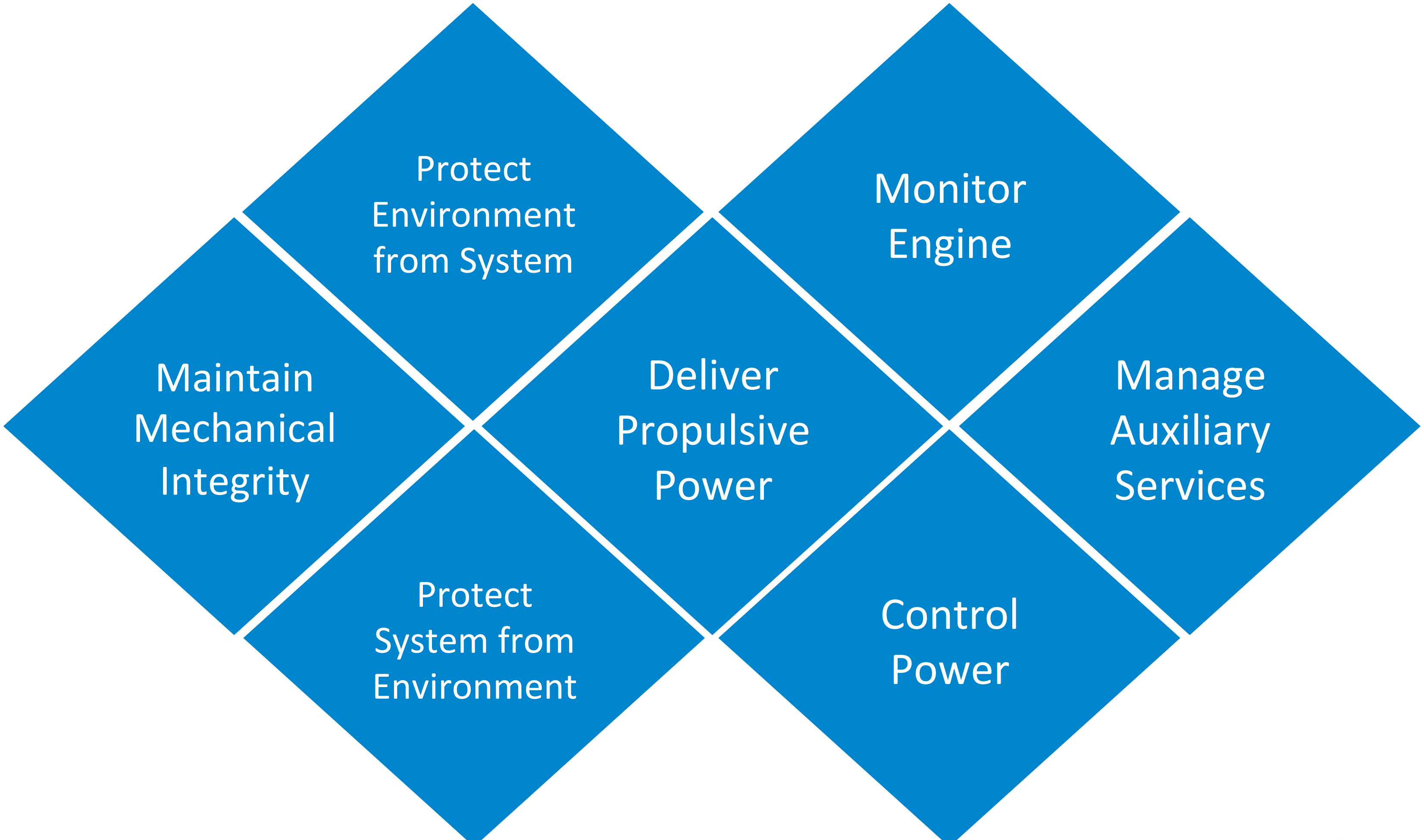
Section	Title	MoSCoW	Subsection	Title	MoSCoW
1	Summary	Must		*(Standard "RR document" content)*	
2	Document Revision History	Must		*(Standard "RR document" content)*	
3	Table of Contents	Must		*(Standard "RR document" content)*	
4	Glossary of Terms	Must		*(Standard "RR document" content Must be consistent with other Project documents)*	
5	Introduction	Must	5.1	Purpose	Must
			5.2	Scope	Must
			5.3	System Context *(Where this system fits in the context of the overall system/ environment)*	Must
			5.4	Document Context	Could

Generic Requirements Document Content (Part 2)

Section	Title	MoSCoW	Subsection	Title	MoSCoW
6	Applicable Requirements	Must	6.1	System Attributes *(Non-Functional System Requirements)*	Must
			6.2	Operating Conditions *(Under which requirements are to be met)*	Must
			6.3	Functional Requirements	Must
			6.3.x	Functional Requirement x	Must
			6.3.x.1	Non-Functional Performance Requirement	Must
			6.3.x.2	Non-Functional Implementation Requirement	Must

Section 6: Every requirement is expressed as:- ID; Requirement (natural language); Rationale/ Discussion (natural language - where from, why, background); Other attributes (as agreed).

Example: Aircraft Gas Turbine Functions



Generic Requirements Document Content (Part 3)

Section	Title	MoSCoW	Subsection	Title	MoSCoW
7	Interfaces	Must	7.1	Functional	Must
			7.2	Physical	Must
8	Trade Factors for use in Trade Studies	Must		*(Comes from "Solution" at the next level up - e.g. unit cost vs. weight)*	
9	Applicable Documents	Could			
10	Appendix (for information only)	Could			

Section 7: Identify each interface (number them); With whom/what the interface is; Who leads; A list of the requirement numbers (from above) that make this interface, or (not preferred) reference an Interface Control Document

4) Definition document

- Captures the response to the requirements for system of interest (so on level)
- Defines (if needed) the sub-elements in the solution, and the derived requirements for them
- Defines (if needed) what is needed from other parts of system for this part to work
  - This is still controversial / difficult, especially in terms of how this is flown to the other parts

Generic Definition Document Content (Part 1)

Section	Title
1	Summary
2	Document Revision History
3	Table of Contents
4	Glossary of Terms
5	Introduction
5.1	Scope
5.1.1	Architecture Overview
5.2	Sub-System Decomposition
6.1	Non-Functional Requirements
6.1.1	Non-Functional Requirement 1
6.1.1.1	Allocation to Sub-system/Component 1
:	:
6.1.1.x	Allocation to Sub-system/Component x
:	:
6.1.y	Non-Functional Requirement y *(same structure as 6.1.1)*
6.2	Operating Conditions (Under which requirements are to be met)

The architecture overview is expected to be developed iteratively as the component and sub-system solutions are refined.

Overview, containing context of the system relative to other systems it interfaces with

Describe chosen system concept, including elements system broken into. Include a "satisfaction argument" – which justifies solution against requirements – explaining how requirements addressed and pedigree of selected solutions

Allocation (and translation), through solution definition, of system attributes to lower level elements

Generic Definition Document Content (Part 2)

Section	Title
6.3	Sub-system functional defintions

	Element 1	Element 2	…	Element m
Function 1				
Fucntion 2				
Function 3				
…..				
Function n				

Breakdown / allocations of system functions to the defined elements.

For each cell break down allocation of solution function (and sub-functions) to each system element. Include

- Derived performance requirement for allocated function
- Any implantation requirements (solution) for functions

Order of sub-sections 6.3.x.y
X = function (keep clear alignment to system functions
Or
X – element – focus on flow to solution elements

Generic Definition Document Content (Part 3)

Section	Title
6.4	Infrastructure
6.5	Interfaces
7	Other derived requirements and "reuqirements not met"

Beyond interfaces a solution may depend on other parts of system / enterprise (i.e **not its sub-elements**)
These are derived requirements for elements not in the "control" of the solution definition.

The mechanism for controlling the flow of this, avoiding "requirements spaghetti, is a challenge.
The needs derived from the solution must be captured

For complex systems, satisfaction of functional requirements implies presence of infrastructure elements. Need to define what the solution requires of them

Top down identification of interfaces created by division into sub-elements

Definition must be clear about requirements that have not been addressed or met

Value of documenting definition

- Clarity of how solution addresses system requirements, and how system is decomposed
- Enables each sub-element to be designed in context, with interfaces identified top down
- Dependencies on other parts of system can be seen
  - Really important in a tightly coupled system
- As solution matures, impacts on the derived requirements for sub-elements can be seen as a whole, at the SOI level

Difficulties encountered

- Seeing all requirements documentation as overhead
- Many, totally disparate, approaches being taken, especially to definition
- Separating Requirements from Definition
  - Especially as requirements documented when solution known
- Flowing requirement to requirement, not through a definition
- How to flow requirements “across structure” –
  - with multiple sources how does integrating system know / control what was asked?
- Separating capturing / analysing requirements from management of requirements (and evidence, definition etc.) information

5) Conclusions

- Requirements, definition and evidence structures/templates have been created that are product domain-neutral but can be tailored to a product domain and to various levels in the system hierarchy
- Need to recognize that in a many level system apply SE to all elements – the challenge is the integration of all the elements
- Interface definitions have been clarified
- Some areas of debate still remain:
  - The use of an “Infrastructure” section in the Definition template
  - How to capture upward flowing requirements
  - How to capture “requirements not met” by the solution definition
- The templates have been tested in the aviation gas turbine environment and have been found to be effective

Thank You!