System Engineering Requirements as part of the Aircraft System Development Process

EASA 13th Rotorcraft Symposium
December 2019

Nicholas D. Kefalas

Platform System Certification Chief / FAA Systems and Equipment DER

LOCKHEED MARTIN

Introduction to Requirements

- Requirements are the starting point of any Design
  - Intended function(s) (Operational and System Performance).
  - Conditional behavior(s) (Output or Operation under specific conditions).
- Requirements should Specify Function. (What a device or system must do ?)
- Requirements should be written with verification in mind rather than design.
- Requirements should be written to be verifiable.

Why Use Requirements?

- If all requirements of a design have been met, a system or device will perform its intended function when installed on an aircraft.

- To demonstrate compliance to Aviation and Safety regulations.

- Provide Engineers and Technicians with a clear set of attributes as to what to design.

- Provide the Verification team with a set of verifiable requirements, that will validate the design and operation.

Types of System Requirements

Customer Requirements	Vary with aircraft type, function of aircraft, and type of system; based on intended payload, routes that will be traveled, operating and maintenance practices, and wanted features
Operational Requirements	Define functions for flight crew, maintenance crew, and support staff; made up of actions, decisions, information, and timing for personnel interactions and interfaces
Performance Requirements	Define what makes system useful (anticipated performance, accuracy, fidelity, range, resolution, speed, response times)
Physical Requirements	Define physical attributes (size, mounting provisions, power, cooling, environmental restrictions, visibility, access, adjustment, handling, storage, and production constraints) and response times

Types of System Requirements (cont.)

Maintenance	Scheduled vs. unscheduled have different types of requirements; some may be needed for certification reasons to meet system safety failure probabilities; requirements define what needs to be done to access and test equipment
Interfaces for Power and Signals	Systems must be compatible with allotted power sources and power budgets; may have impedance and electrical characteristics requirements for operating properly; these requirements define how signals designated as system outputs in the interface specifications are created and how signals designated as system inputs in the interface specification are used
Certification Process	Regulations or authorities define requirements of features, attributes, or specific implementations; typically to show compliance with airworthiness regulations

System Requirements

- Safety and Failure Analysis is the basis of the Life Cycle Development Process of any system or device. Regulated by processes and activities such as those described in ARP4754A and ARP4761.

- Performance constraints associated with the integrity of function originate from the system safety process. Safety specific requirements should be identified so they can be traced from design to implementation.

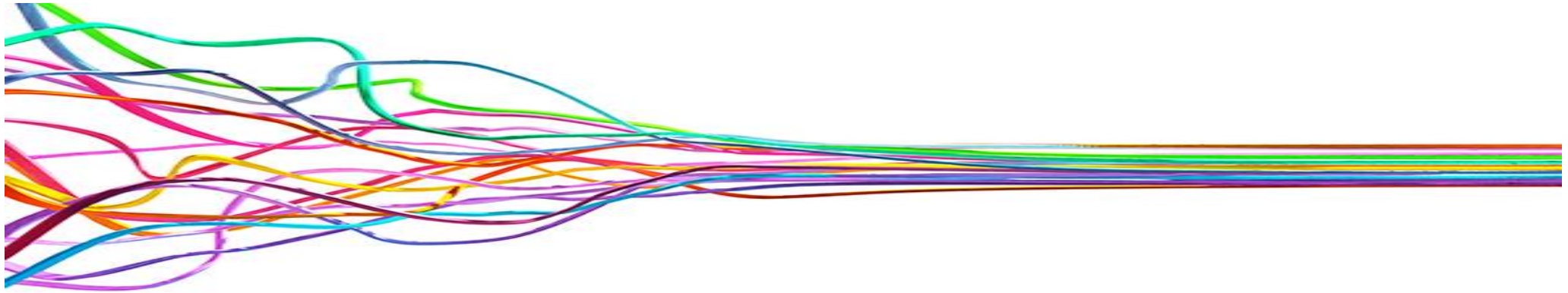
- Gathered by system engineering development process.
  - Desired performance of system under operating conditions
  - Function of system or device based on customer desires, operational constraints, regulatory restrictions, and implementation realities

Creating requirements…(The Challenges)

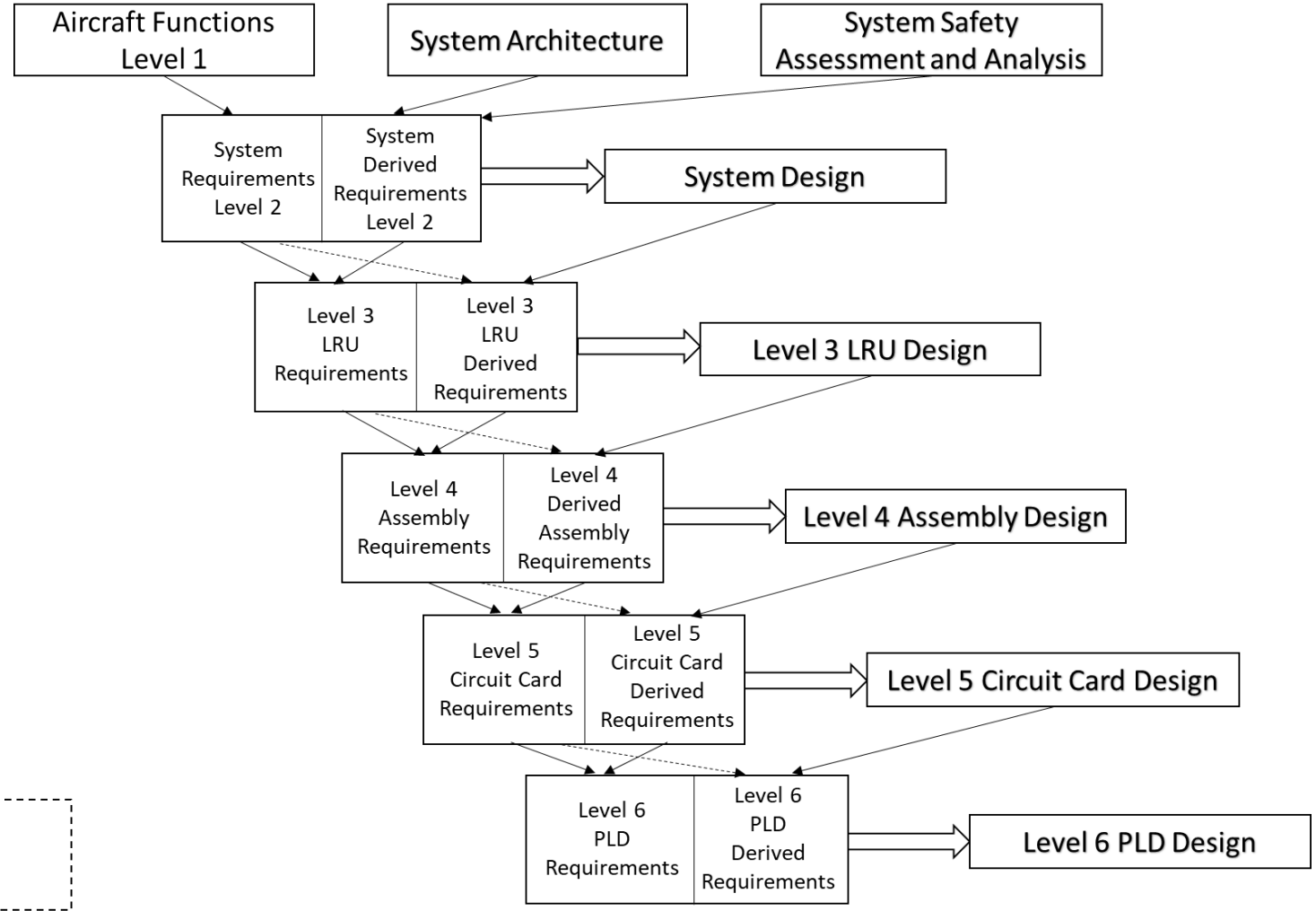
- The ability to write about functionality without including design or implementation details. (Paradox)
- Knowledge of the intended function and operational requirements.
- Design decisions for the next decomposition level, which will in turn define/affect the requirements at that next tier.
- Describing function
- Describing design

After Gathering Requirements…

- All requirements sorted into categories (physical, environmental, performance, functional, etc.)
- Requirements are validated though review, analysis, and testing to determine whether they are correct/ complete.
- Additional Requirement added/removed/altered based on design decisions

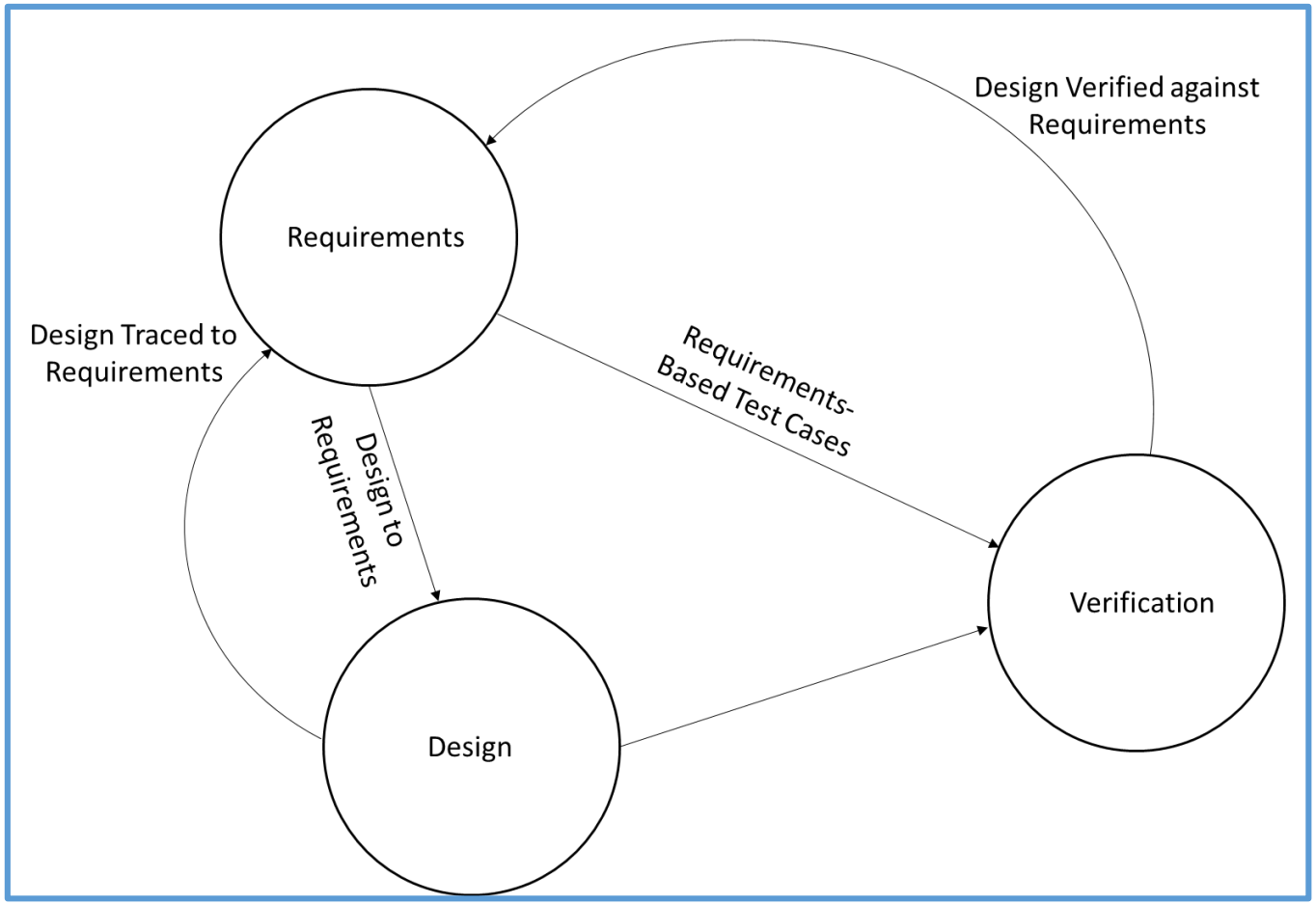


Decomposition of Functional Requirements Example

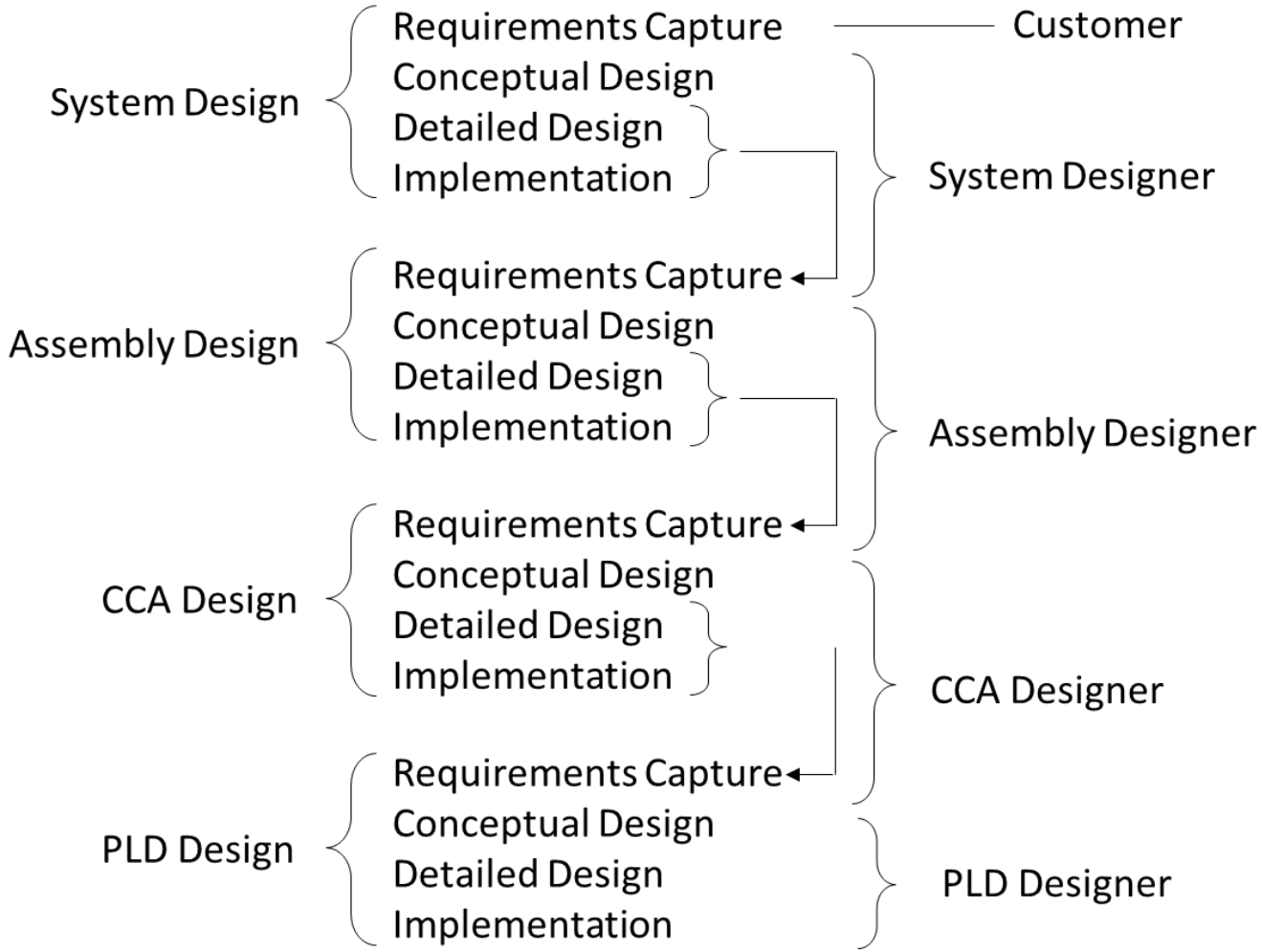


Note:
LRU- Line replaceable unit

The Traceability Game



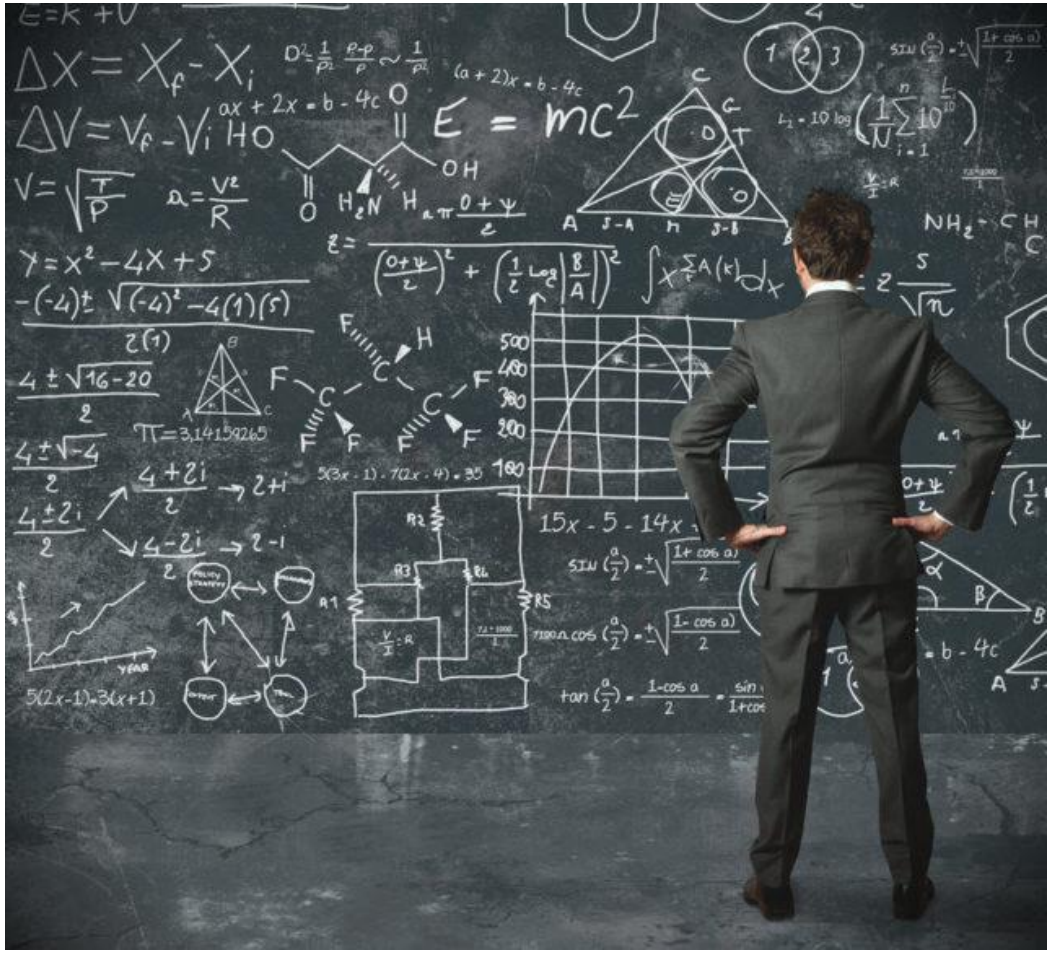
Requirements Capture Example (Electronic)



Note:
CCA- Circuit card assemblies
PLD- Programmable logic devices

Types of Requirements for Typical Systems

- Functional requirements
- Application requirements
- Implementation requirements
- Input requirements
- Indirect requirements
- Referred requirements
- Global requirements
- Derived requirements



Requirements Types Explained

Functional Requirements

- Describe the behavior of a function. In example, “how” an output responds to an input or stimulus.

Application Requirements

- Describe the “How” hardware will be used

Implementation Requirements

- These type of requirements describe aspects of hardware design and/or implementation

Input Requirements

- Express a descriptive text or in interface specification (not typically used. Subjective in nature).

Indirect Requirements

- Requirements that imply functionality.

Requirements Types Explained (Cont …)

Referred Requirements

- Requirements that have external references or inter system associations.

Global Requirements

- Requirements that describe response or action driven by an event or input to a group or set of events or outputs.

Derived Requirements

- Requirements that are typically “off springs” of higher level requirements or are the product of design decisions.

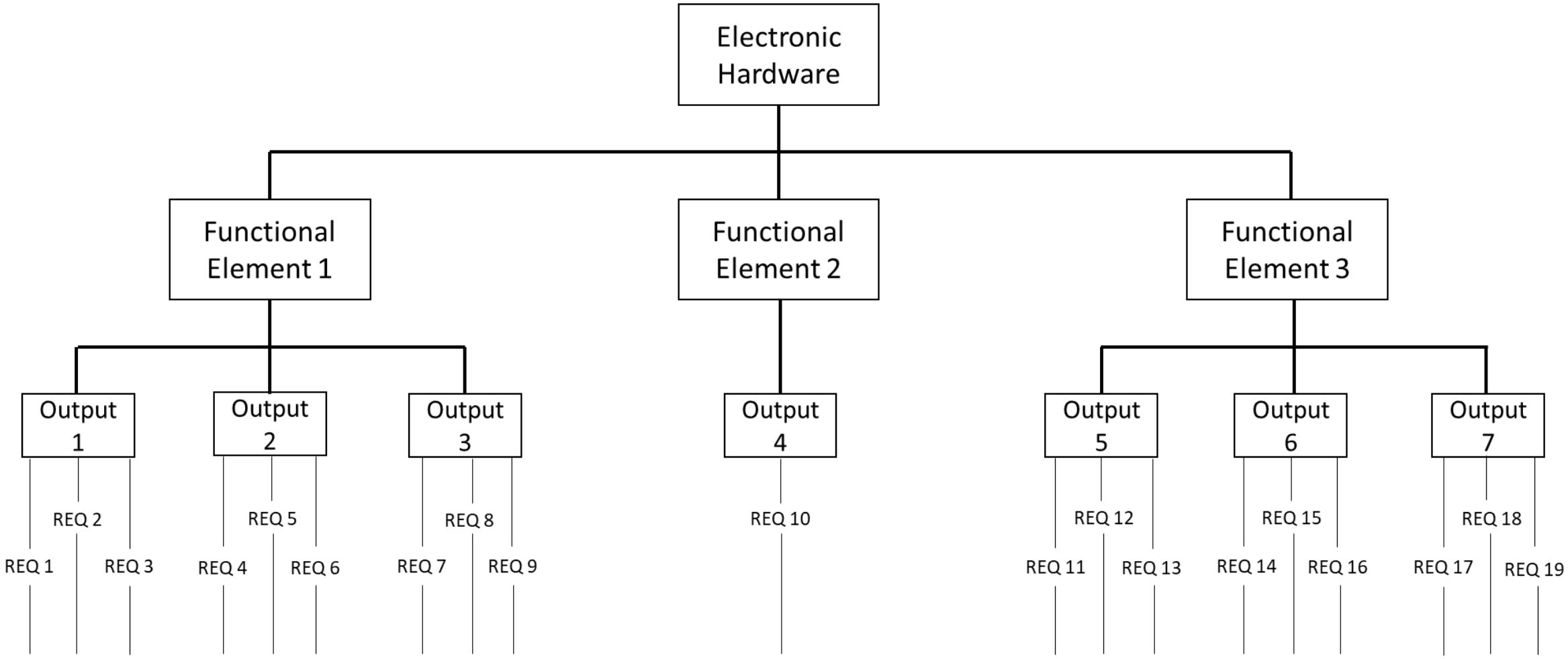
Allocation and Decomposition

- Functions become requirements for systems or devices by decomposing and assigning them to different sub systems or elements of the device

- Simple systems may use system requirements flowing directly to hardware requirements

- Complex systems (highly integrated functionality)
  - Decomposition begins with system requirements and allocate through the different levels of design in system
  - Functions get allocated or decomposed from higher level to lower level requirements

Requirements Organization Layout



Writing Requirements Guidelines

- Use keyword “shall” to identify requirement (not always needed)
- Assign unique identifier to each requirement
- Describe *what* is done, not *how* it is done
- Explain how outputs respond to valid inputs, invalid inputs, and timing related events
- Be specific and concise in each requirement
- Positive sense, no “shall not” requirements (difficult to prove that something *does not* happen)
- Make each requirement unique
- Express requirements in terms of behavior
- Use descriptions of outputs that can be observed and descriptions of inputs that can be controlled
- Write for target audience—design engineer and verification engineer

Standard Form for Writing Requirements

- Enhances requirements capture and interpretation
- Easier to find information in requirements document, so team members can be more ready to contribute
- Ensures consistency and simple for new employees to catch on
- Typical template/form:

Function Name

Definition of terminology

Trace links

Justification for derived requirements

Output(s)

- Description
- Units
- Encoding (bus, number of bits and weighting of least significant digit)

Input(s) affecting output

Power-on behavior—behavior of signal when power first applied

Reset response—state of signal after reset asserts or deasserts

Assert behavior—requirements for asserting output to its active state

Deassert behavior—requirements for deasserting output from its active state

Invalid behavior—how signal should behave when presented with invalid or undefined inputs

Requirement Considerations in Systems

System design can capture the architecture and features necessary for implementing and realizing the system functionality, safety, and reliability

Aircraft level requirements can be allocated to systems

Software requirements can be allocated/ decomposed from the requirements for the circuit card that hosts the microprocessor or microcontroller

Aircraft functions can be identified and expressed as requirements

System requirements can capture the system functionality and any additional requirements from the safety processes

Design can show how requirements are decomposed

Allocation and decomposition can continue down through one or more levels of abstraction

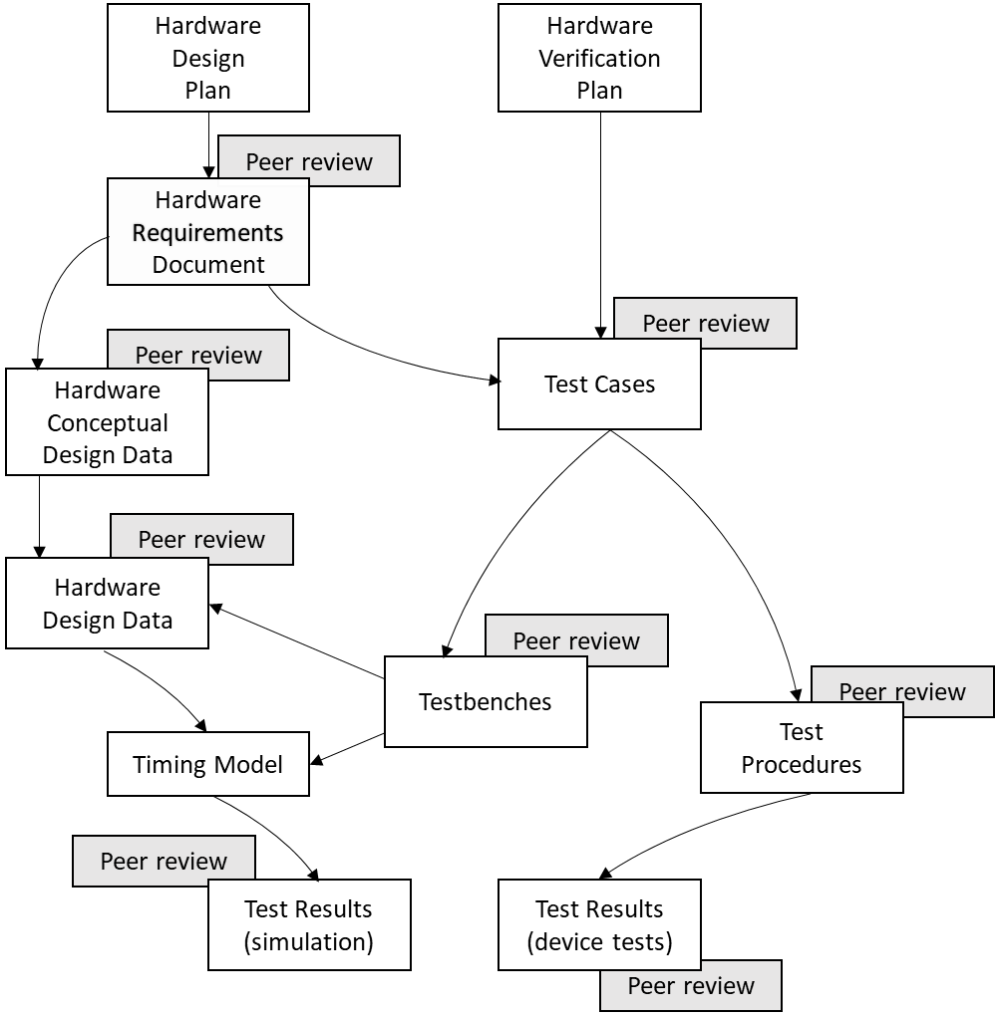
Each level of abstraction can repeat its allocated requirements until there is a design

PLD requirements can be allocated/ decomposed from the requirements for the circuit card that hosts the PLD device

Introduction to Verification

- Verification is performed to assure compliance to all system requirements.
- Hardware and Software is evaluated against its requirements to demonstrate that it performs intended function
- Verification activities include Inspection, Analysis, Simulation and Test
  - Inspection- a qualitative assessment to verify a drawing or document or data against a requirement or group of requirements.
  - Analysis- a quantitative and qualitative (can be both) assessment of a requirement or group of requirements against measurable standards or data / results that can be correlated to said requirement(s).
  - Simulation – quantitative testing that can provide functionally equivalent results of the requirement or set of requirements by mimicking in aircraft operational environments.
  - Testing- only method by which behavior of actual requirements and/or device and/or system can be observed and quantitatively measured.

Example of Verification Structure for a Hardware Development Life Cycle



Note:
testbench- synonym for a test procedure used in simulation

Example of a Hardware Test Matrix

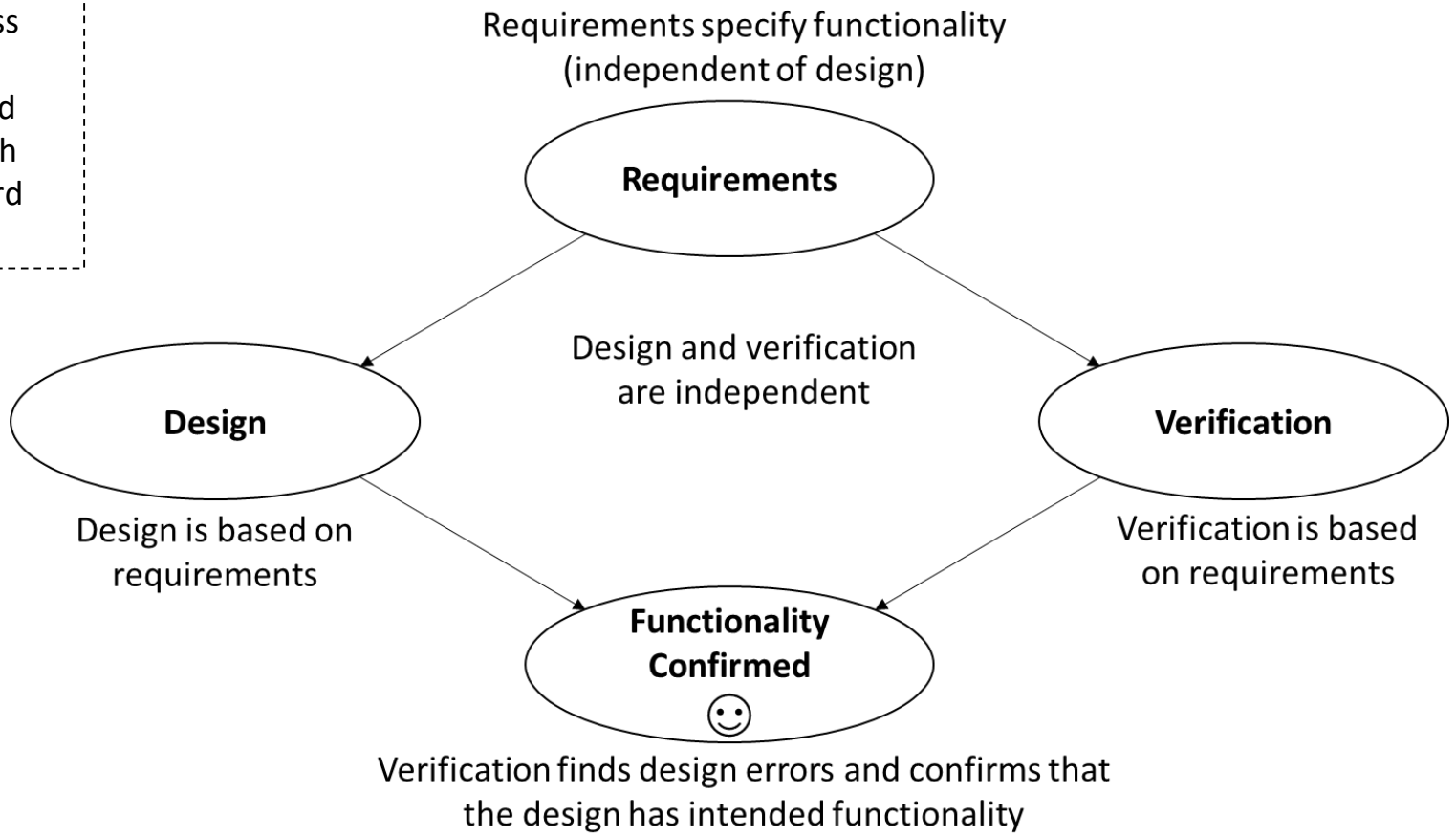
Note:
HW- Hardware
SW- Software

Requirement ID	Review	Analysis	Simulation		In-Circuit Test	
			Functional	Post-Layout Timing	Circuit Card Tests	HW/SW Integration
Functional Element 023 (UART)						
HRD-ABC-123-001	HRD review	Dynamic timing	X	X	X	
HRD-ABC-123-002	HRD review	Dynamic timing	X	X		X
HRD-ABC-123-003	HRD review	Static timing	X			X
HRD-ABC-123-004	HRD review	Dynamic & Static timing	X	X		X
HRD-ABC-123-005	Inspection	Static timing	X		X	X

Functional Requirements Effect on Verification

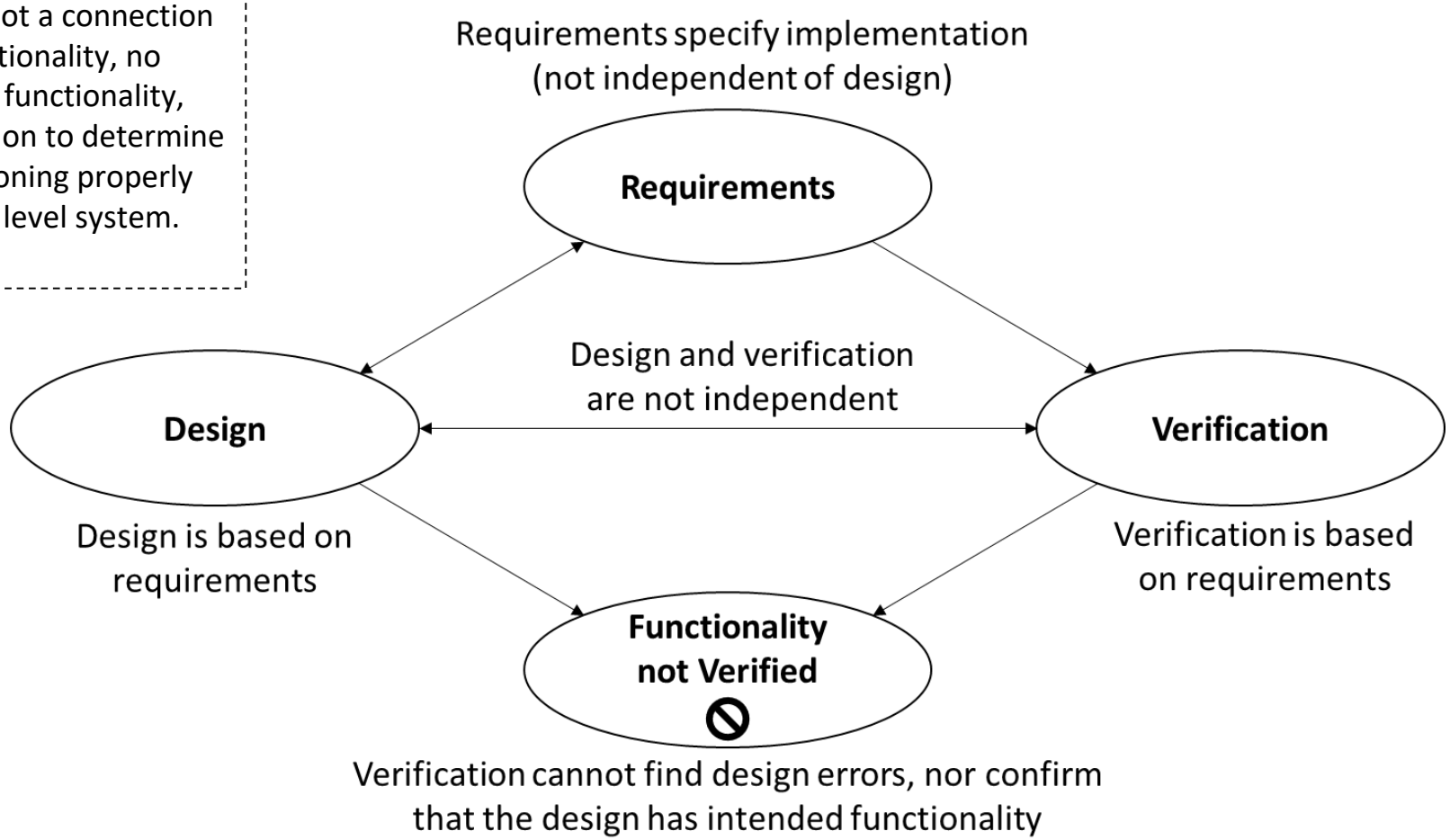
Requirements must state *what* the hardware does (functionality), not *how* it should do it (implementation)… These following figures show the effects on integrity of the verification process if design information is in requirements.

Requirements that express functionality are equally appropriate for design and verification and allow both processes to move forward independently.



Implementation Requirements Effect on Verification

When design information gets into requirements, there is not a connection to the system level functionality, no capture of the intended functionality, and no way for verification to determine that the design is functioning properly as defined by the upper level system.



Product Service Experience

- When used, must be coordinated with certification authorities and stated in the PHAC and must document current and/or previous use of hardware
- Service history should be based on the same component with the same part number and version
- Service experience from flight test programs are not suitable for demonstrating compliance to DO-178C / DO-254, since the aircraft is not yet certified

QUESTIONS

THANK YOU