ELECTRONICDAYS 2012

Objectives of the training

- Introduction
- Presentation of the DO-254
- Complexity considerations
- Hardware planning process
- Hardware design processes
- Supporting processes

INTRODUCTION
Different root causes of failures

Different root causes

- Random failure
- Error
- Event

Random Failure

Example: short circuit of electronic component

- This type of failure is covered by « classical » safety analyses such as FMEA, FMES, FTA,…

Error

Human error: Example: human error on procedure application

=> Human error are covered by Human Factor analyses and completed by ZSA (Zonal Safety Analysis)

Design/development errors: Examples : specification error, hardware/software design error.

=> Introduction of ***DAL*** (Design Assurance Level) to reduce the risk of error

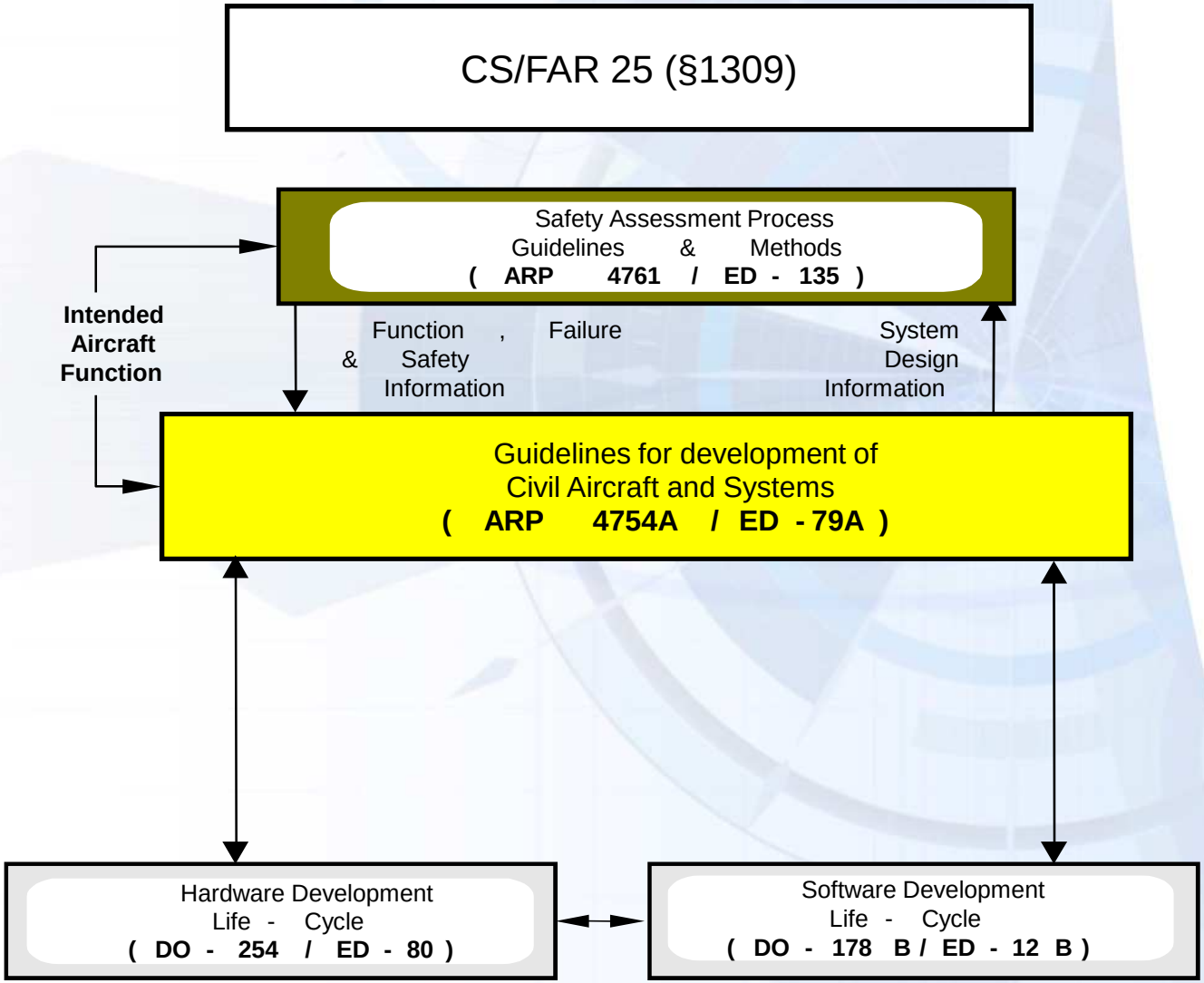
Event

- Examples of events considered in safety analysis:
  - **Environmental events: meteorological conditions**
    - Wind, Icing, lightning, etc.
  - **Flight Operation**
    - RTO, terminal area, emergency evacuation.
  - **Fire or smoke event**
    - Fire or smoke in the cockpit or in cabine (APU, Engines,…)
  - **Bird impact**

Global approach of safety

- *Randum failure* → « classical » safety analysis
- *Error (human cause)* → *Human factor analysis*
- *Error (design cause)* → *Design Assurance Level*
- *Event* → *to be dealt in SSA and PRA analysis*

Introduction



ED79-ARP4754

Presentation of ARP4754

ARP4761

ARP4754

Function 1 (eg: provide electrical power)

Function 2 (eg: propulsion)

Function n

System 1 (ex: electrical distribution)

System 2 (ex: electrical generation)

System 3

System 4

System n

Item 1 (ex: CPIOM calculator)

Item 2

Item n

DO178

DO254

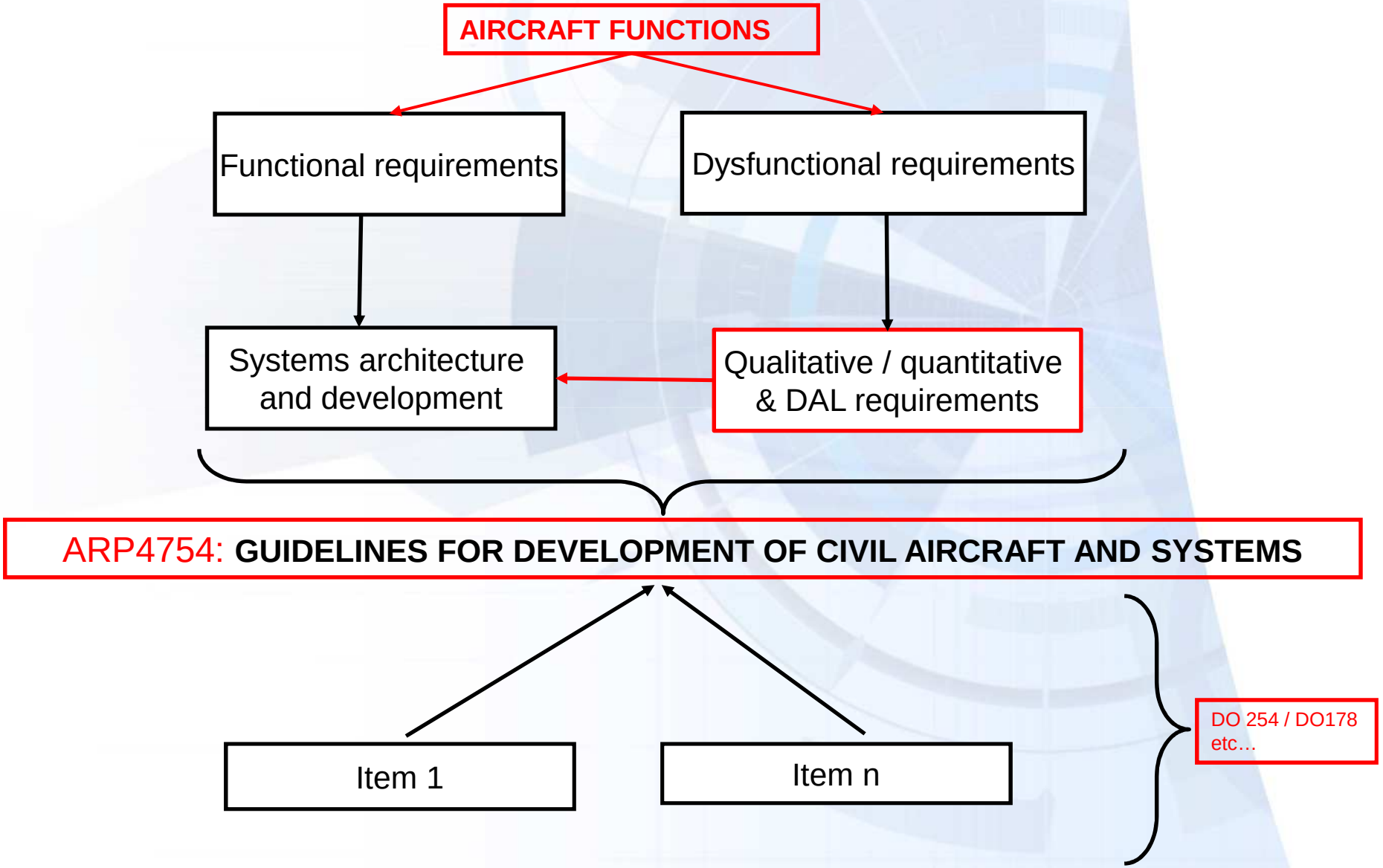
Hardware 1 (ex: FPGA, assembly board)

Hardware n

Software 1 (ex: boot, loader, applications)

Software n

Main purpose of ARP4754



SAFETY ASSESSMENT PROCESS

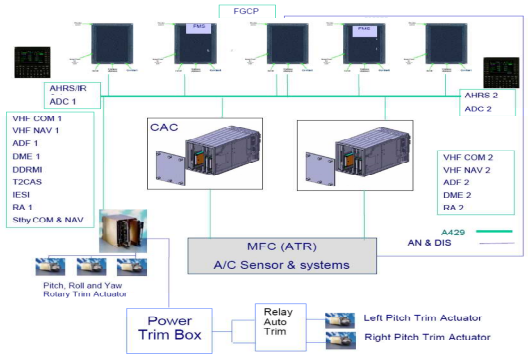
Aircraft Functions

Aircraft level

Function DAL assigment

FHA

SyFHA



Item DAL allocation

Quantitative safety requirements

PSSA

System level

FTA

FMEA

CMA

FMES

Hardware level

Objectives of the training

- Introduction
- Presentation of the DO-254
- Complexity considerations
- Hardware planning process
- Hardware design processes
- Supporting processes

Design of Airborne Electronic Hardware

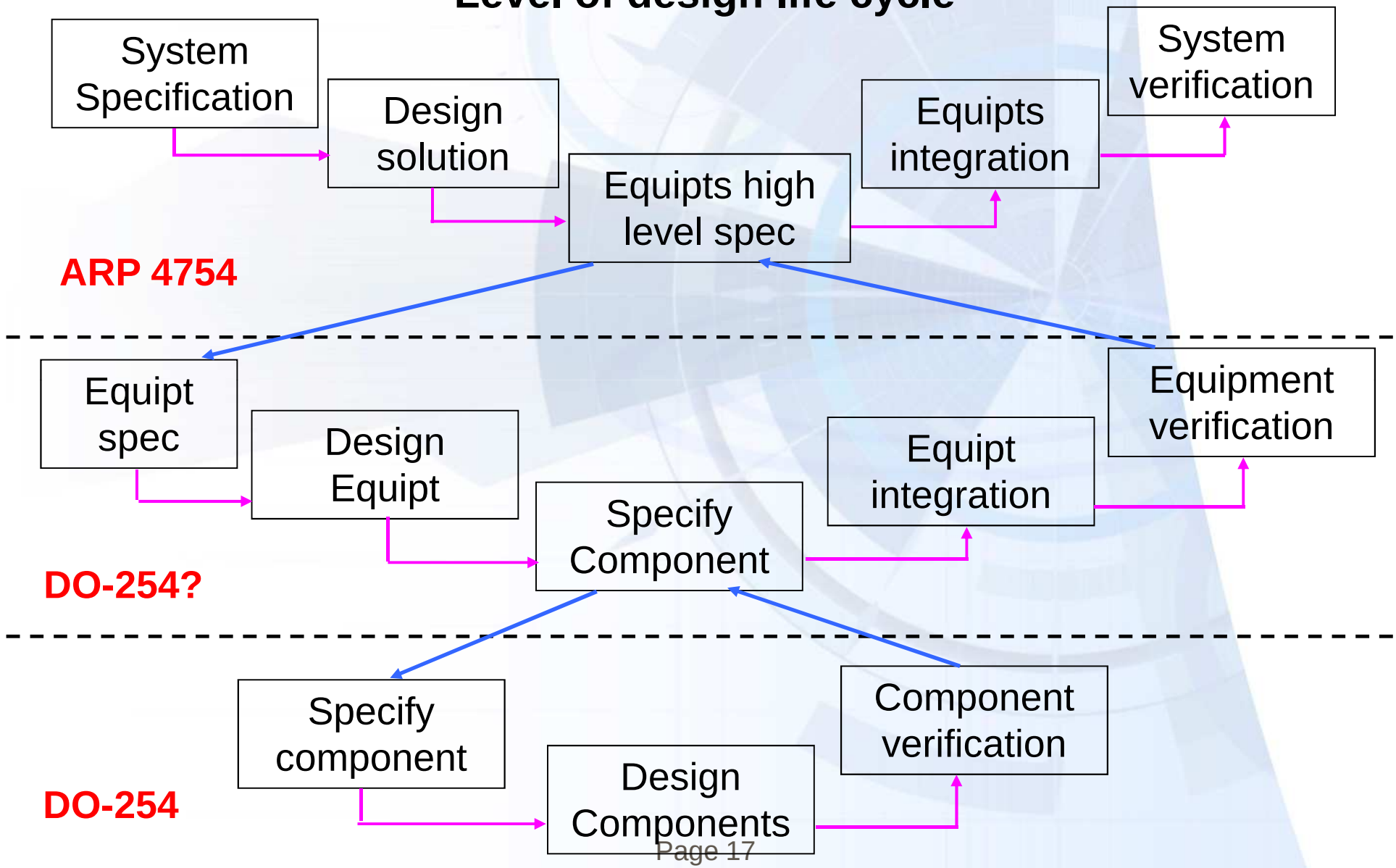
ED80-DO254

Presentation of DO-254 (3/12)

- The use of integrated functions increases the complexity
- The risk of design error increases
- The design rules are not adequate to prevent all potential design errors and therefore to satisfy the specification with a good level of confidence

Need to avoid or detect errors by development assurance activities

Presentation of DO-254 (7/12)
Level of design life cycle



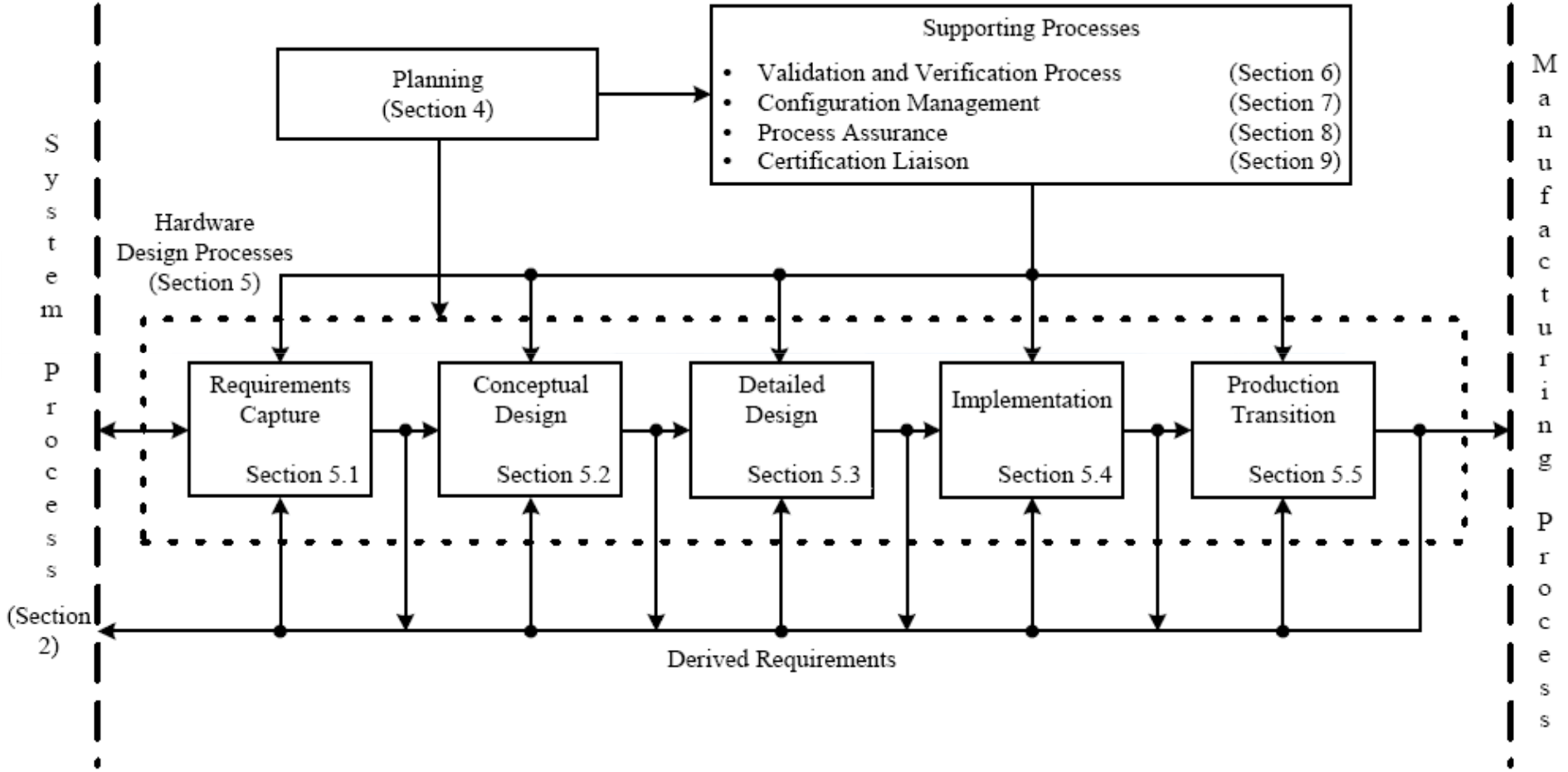
Presentation of DO-254 (8/12)

- RTCA DO-254/ EUROCAE ED-80 is designed to fill the gap for developmental assurance for complex electronic hardware including:
  - Line Replaceable Units (LRUs)
  - Circuit Board Assemblies
  - Custom micro-coded components such as ASICs, PLDs, FPGAs, including any associated macro functions.
  - Integrated technology components, such as hybrids and multi-chip modules
  - Commercial-Off-The-Shelf (COTS) hardware components

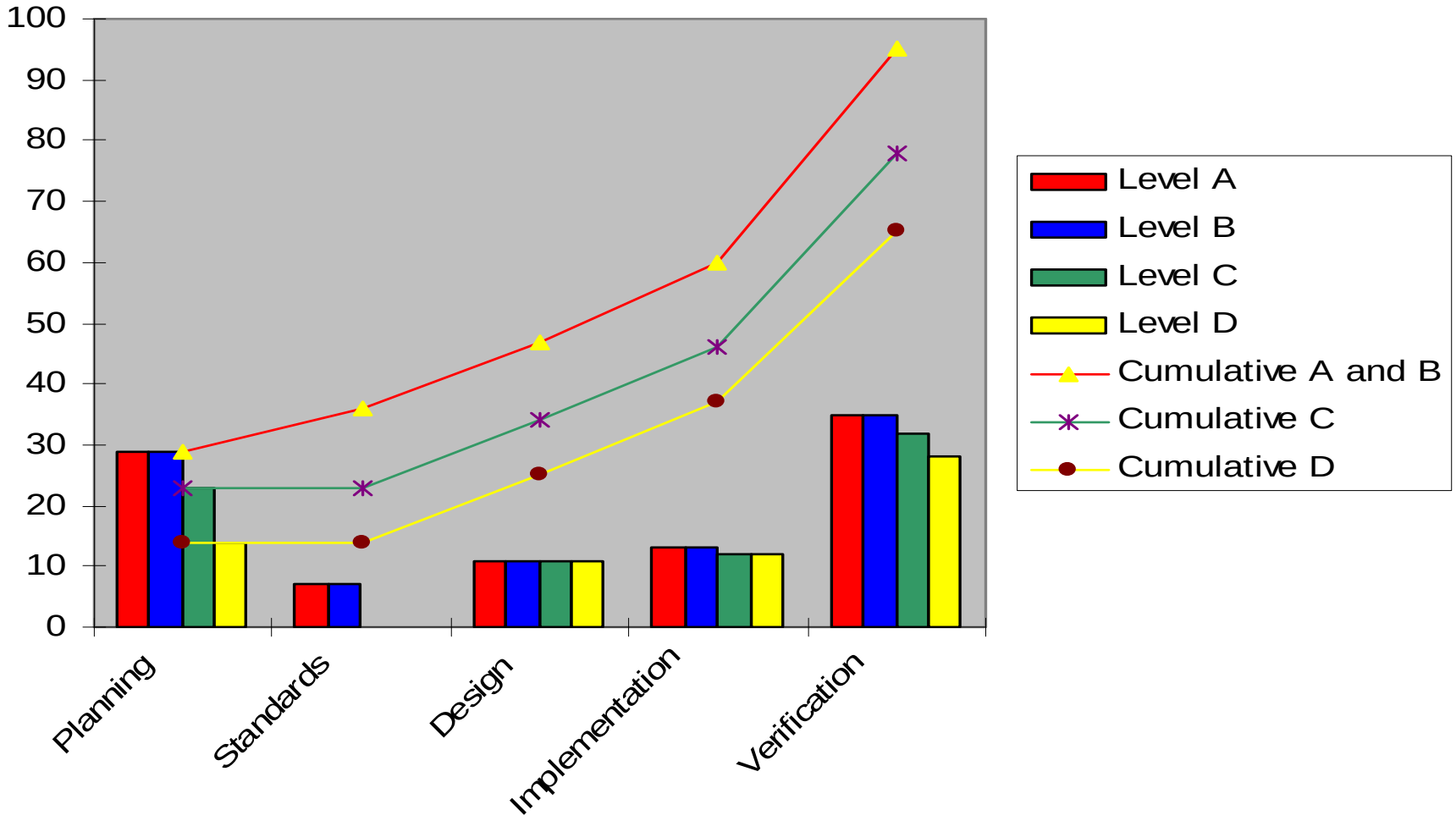
Presentation of DO-254 (9/12)
DO-254 limitations

- AC 20-152 was released at the end of June (2005).
- The AC alters the application of DO-254 in fundamental ways:
  - Scope is limited to complex devices ONLY – application of DO-254 to complete Line Replaceable Units (LRUs) and Circuit Card Assemblies (CCAs) is no longer required
  - Application at level D is optional and not subject to FAA oversight/approval
  - Strengthens exemption for Commercial-Off-The-Shelf (COTS) microprocessors
- The AC is applicable for every form of certification [e.g., Type Certification (TC), Technical Standard Order (TSO) authorizations, Parts Manufacturer Approval (PMA)]

Presentation of DO-254 (11/12)
DO-254 Content



Hardware design lifecycle
DESIGN ASSURANCE WORKLOAD (1/4)

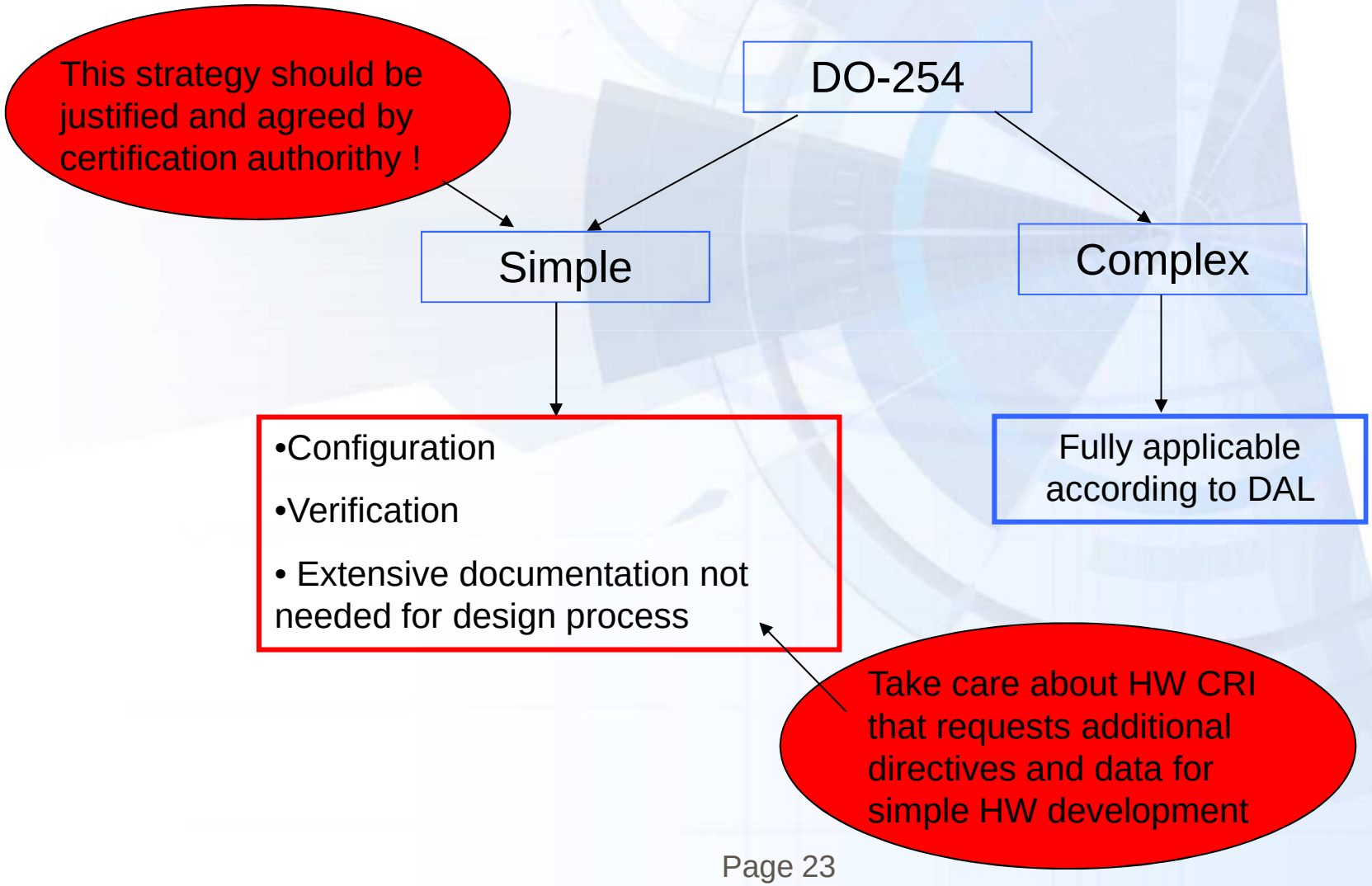


Objectives of the training

- Introduction
- Presentation of the DO-254
- Complexity considerations
- Hardware planning process
- Hardware design processes
- Supporting processes

Complexity Consideration (3/3)

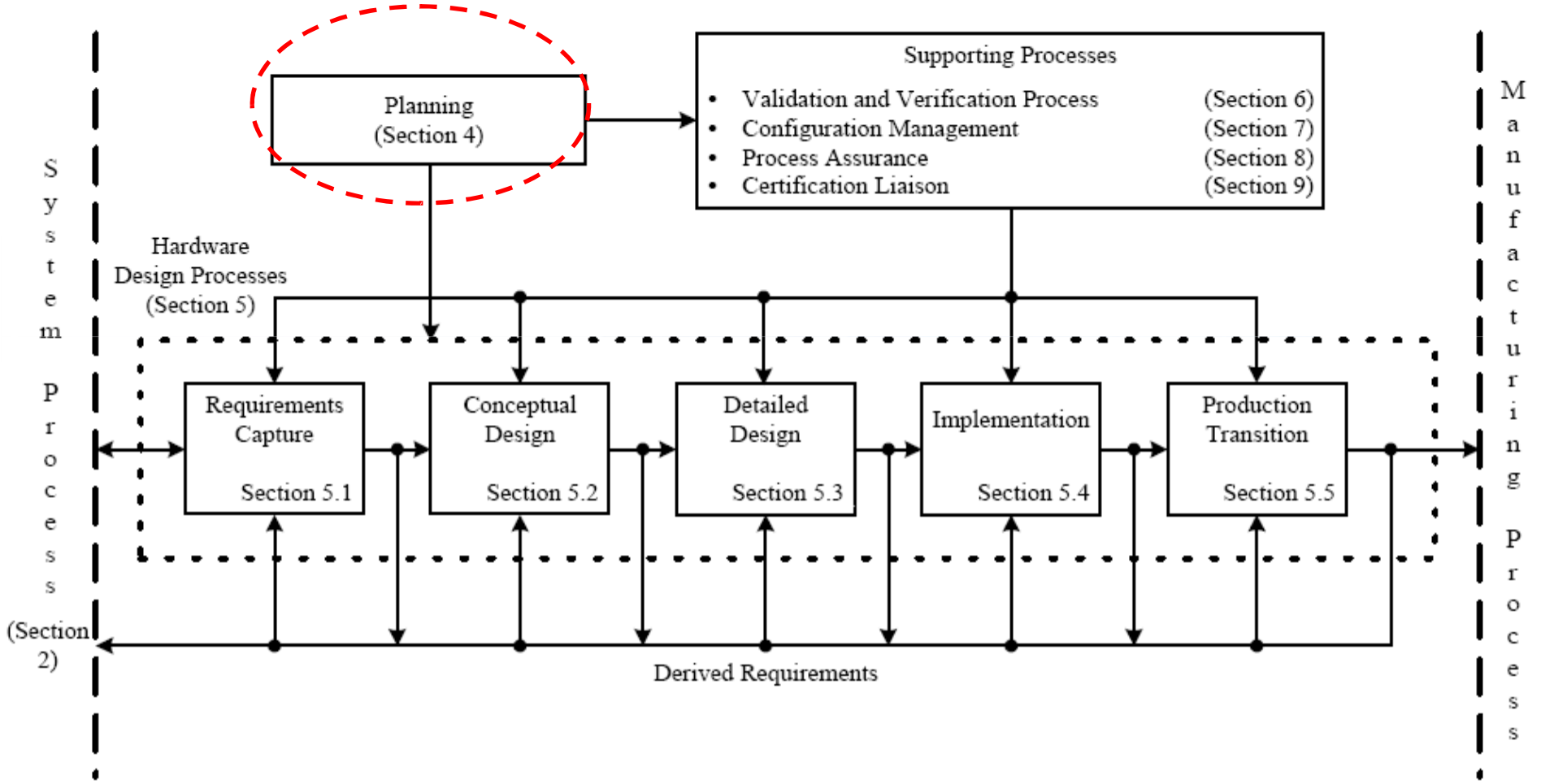
Activities:



Objectives of the training

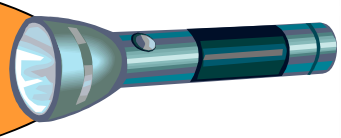
- Introduction
- Presentation of the DO-254
- Complexity considerations
- Hardware planning process
- Hardware design processes
- Supporting processes

Hardware planning process (4/4)



Hardware planning process (1/4)

PLANNING PROCESS



DIRECT PROCESSES:

- Requirement capture
- Design
- Implemention
- Production

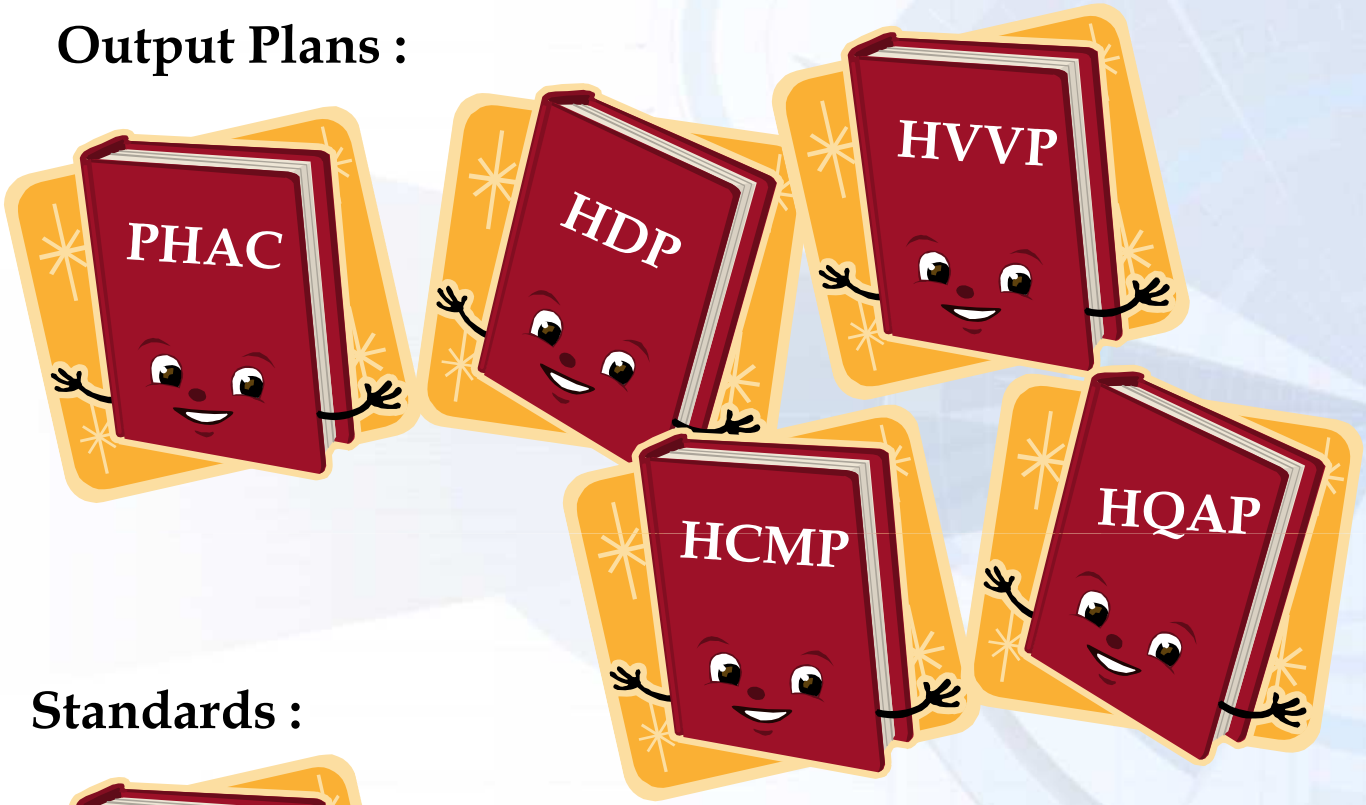
INTEGRAL PROCESSES:

- Verification
- Configuration management
- Process assurance
- Certification liaison

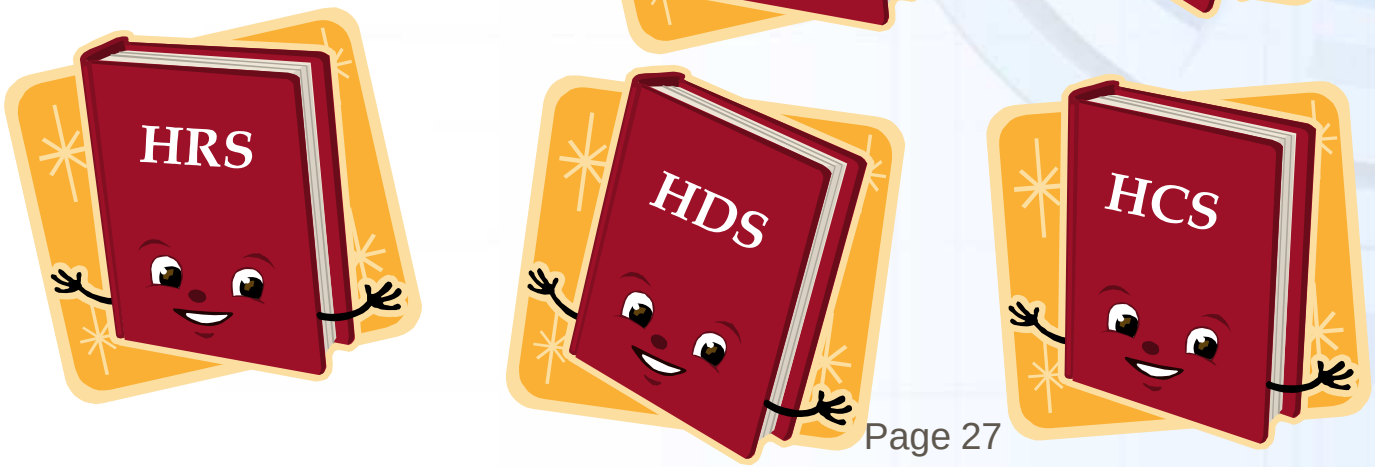
ADDITIONAL CONSIDERATIONS

Hardware planning process (3/4)

Output Plans :



Standards :

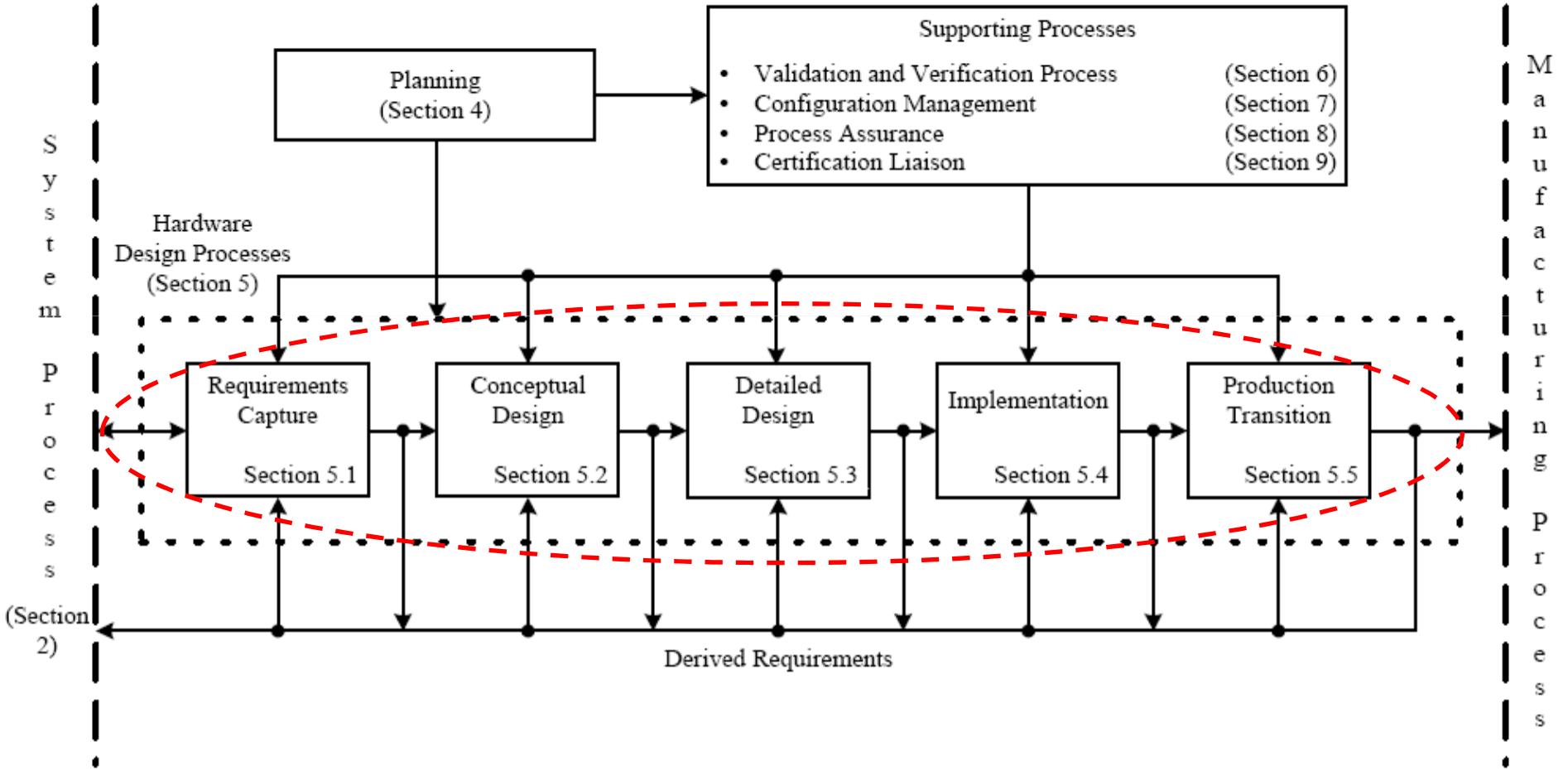


Required for levels A, B, C, D except for HQAP (A,B), HDP (A,B,C) & HVP (validation) (A,B,C)

Objectives of the training

- Presentation of the DO-254
- Complexity considerations
- Hardware planning process
- Hardware design processes
- Supporting processes

Hardware Design Process (2/20)



Requirements capture

Hardware requirement capture
What is a requirement?

DO-254 definition*:* *An identifiable element of a specification that is verifiable*

- A requirement defines a need (WHAT), not a solution (HOW)
- A requirement must be identified by:
  - Req_id
  - Using word as:
    - shall,
    - should,
    - may,
    - must

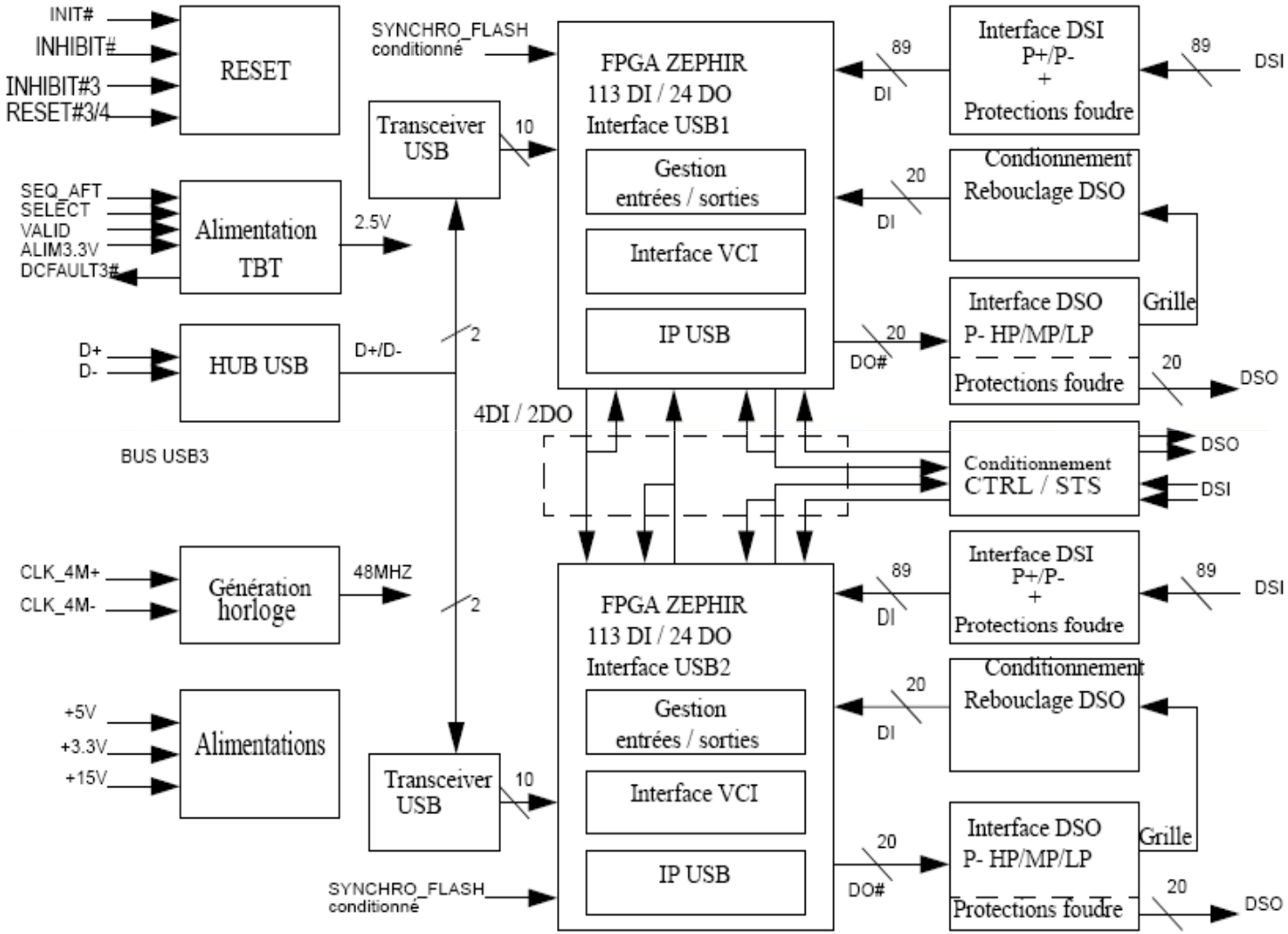
Hardware requirement capture
Type of requirements

The requirements can be classified into 3 categories:

- **Direct** => Requirement directly cascaded from upper specification without rewording
- **Refined** => requirement linked to upper requirement but reworded and detailed. It could be divided into several requirements.
- **Derived**

Conceptual Design

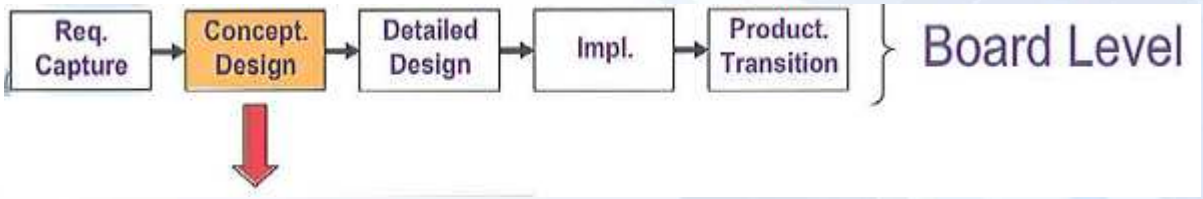
Conceptual Design example (10/20)

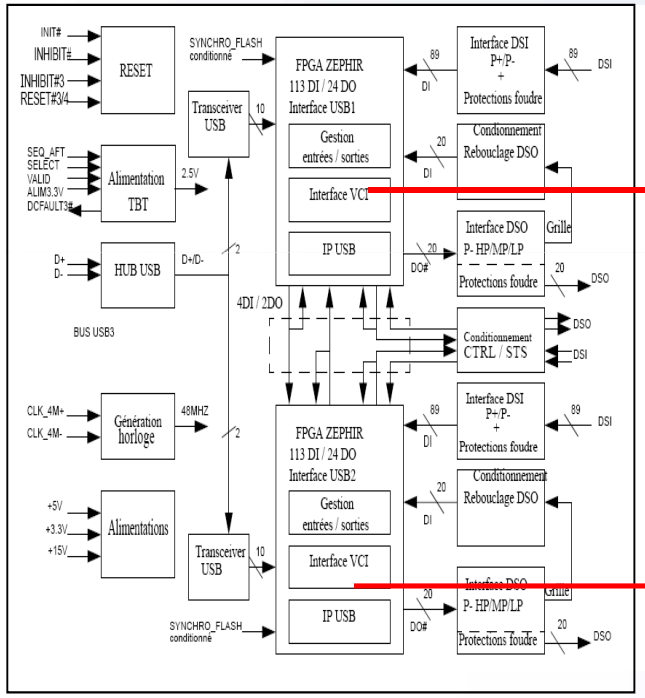


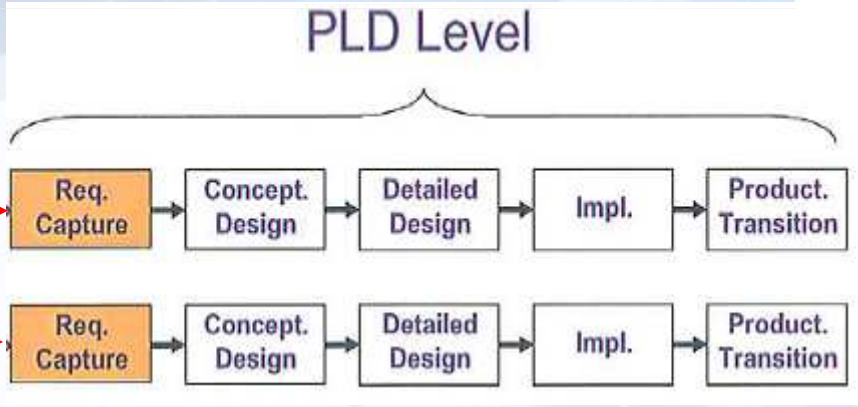
Hardware conceptual example (12/20)

```
library IEEE;
use IEEE.numeric_std.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.std_logic_1164.all;
use STD.textio.all;

use work.pkg_ioconf.all;
use work.pkg_print.all;
use work.pkg_comp.all;
use work.pkg_sxga.all;


architecture normal of driver is

 constant PERIOD           : time := 12 ns;
 signal cnt              : integer := 0;
 signal file_name_read_1         : string(1 to 10) := "image3.pgm";
 signal file_name_written_1      : string(1 to 12) := "imgread3.pgm";
 signal file_name_read_2         : string(1 to 10) := "image4.pgm";
 signal file_name_written_2      : string(1 to 12) := "imgread4.pgm";
 signal file_name_written_3      : string(1 to 12) := "imgread5.pgm";
 signal file_name_written_4    : string(1 to 12) := "imgread6.pgm";
 signal lvl_dis                : log_level := ERROR;
 signal name                   : string(1 to 12) := "SC_NORMAL    ";
 signal check_error_status     : boolean := false;
 signal type_verif             : integer:=0;
 signal check_ok         : std_logic;
```

Conceptual Design example (13/20)

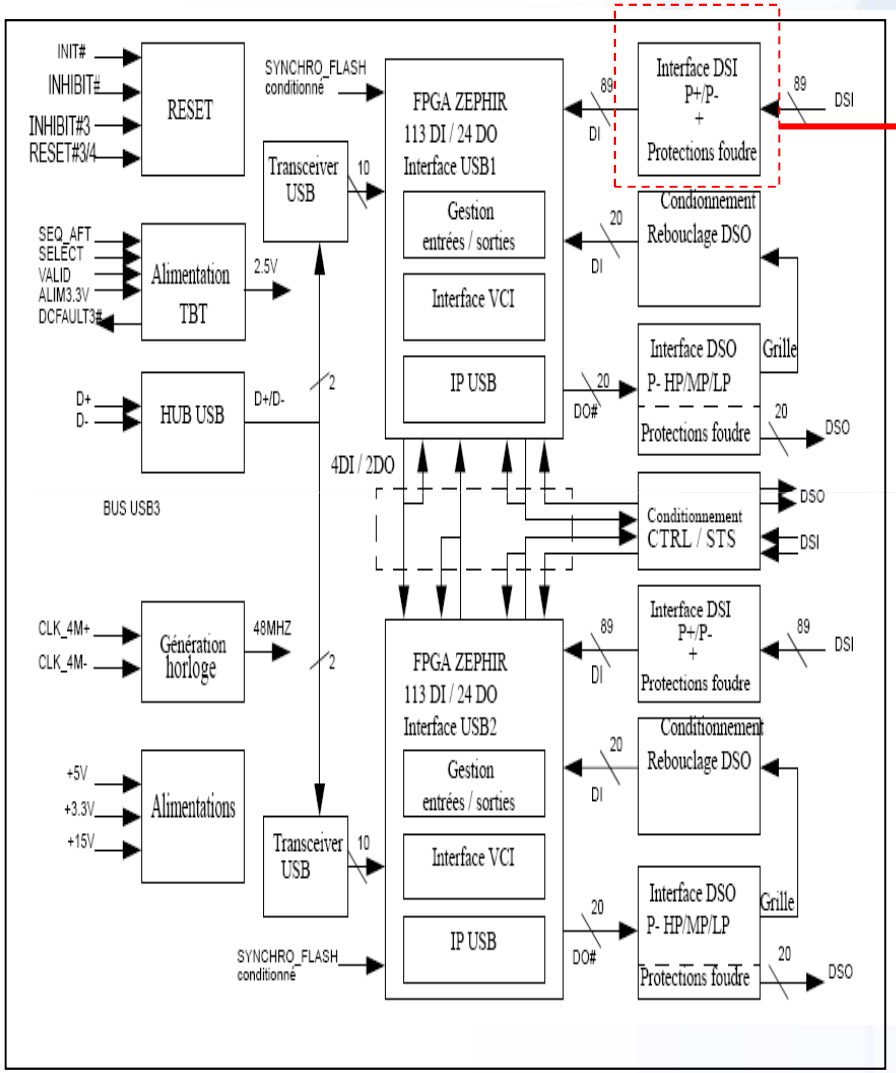




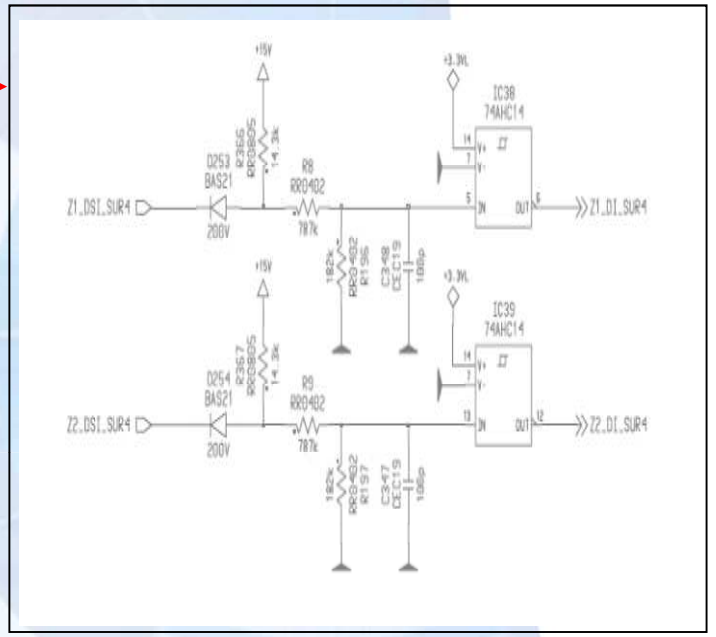


Detailed Design

Hardware Detailed Design exemple (15/20)



Conceptual design



Detailed design

Hardware Detailed Design exemple (16/20)

```
library IEEE;
use IEEE.numeric_std.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.std_logic_1164.all;
use STD.textio.all;

use work.pkg_ioconf.all;
use work.pkg_print.all;
use work.pkg_comp.all;
use work.pkg_sxga.all;


architecture normal of driver is

 constant PERIOD           : time := 12 ns;
 signal cnt              : integer := 0;
 signal file_name_read_1         : string(1 to 10) := "image3.pgm";
 signal file_name_written_1      : string(1 to 12) := "imgread3.pgm";
 signal file_name_read_2         : string(1 to 10) := "image4.pgm";
 signal file_name_written_2      : string(1 to 12) := "imgread4.pgm";
 signal file_name_written_3      : string(1 to 12) := "imgread5.pgm";
 signal file_name_written_4    : string(1 to 12) := "imgread6.pgm";
 signal lvl_dis                : log_level := ERROR;
 signal name                   : string(1 to 12) := "SC_NORMAL   ";
 signal check_error_status     : boolean := false;
 signal type_verif             : integer:=0;
 signal check_ok         : std_logic;
```

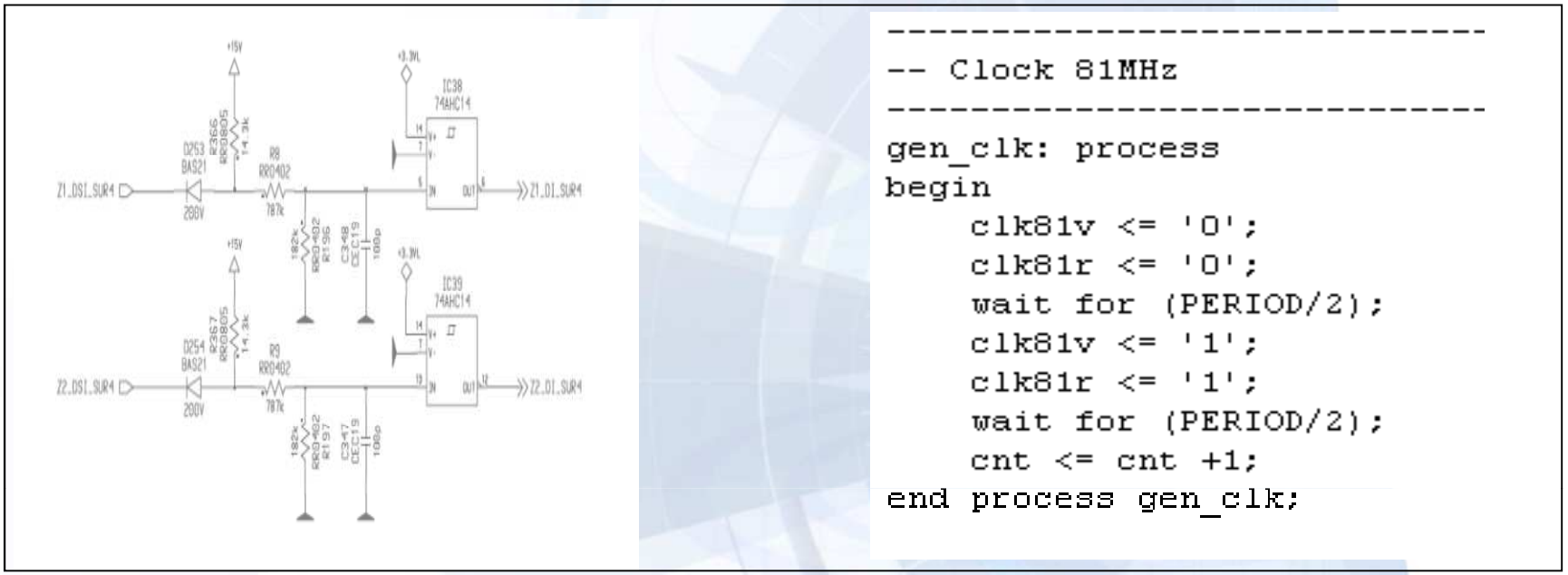
Conceptual design

```
------------------------------
-- Clock 81MHz
------------------------------
gen_clk: process
begin
     clk81v <= '0';
     clk81r <= '0';
     wait for (PERIOD/2);
     clk81v <= '1';
     clk81r <= '1';
     wait for (PERIOD/2);
     cnt <= cnt +1;
end process gen_clk;
```

Detailed design

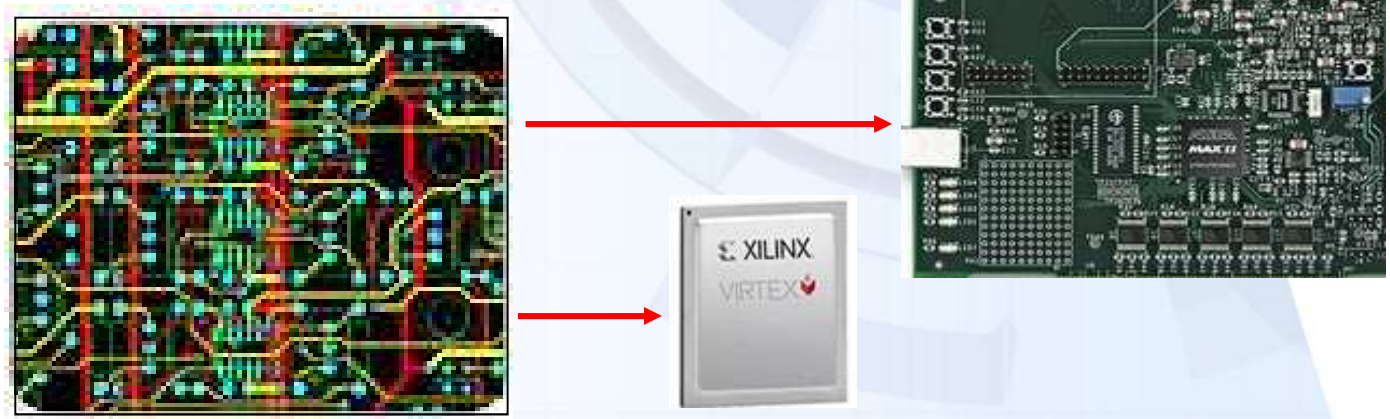
Implementation

Hardware Implementation (18/20)



```
------------------------------
-- Clock 81MHz
------------------------------
gen_clk: process
begin
     clk81v <= '0';
     clk81r <= '0';
     wait for (PERIOD/2);
     clk81v <= '1';
     clk81r <= '1';
     wait for (PERIOD/2);
     cnt <= cnt +1;
end process gen_clk;
```

Detailed design



Implementation

Production Transition

Hardware Production Transition (19/20)

Objectives:

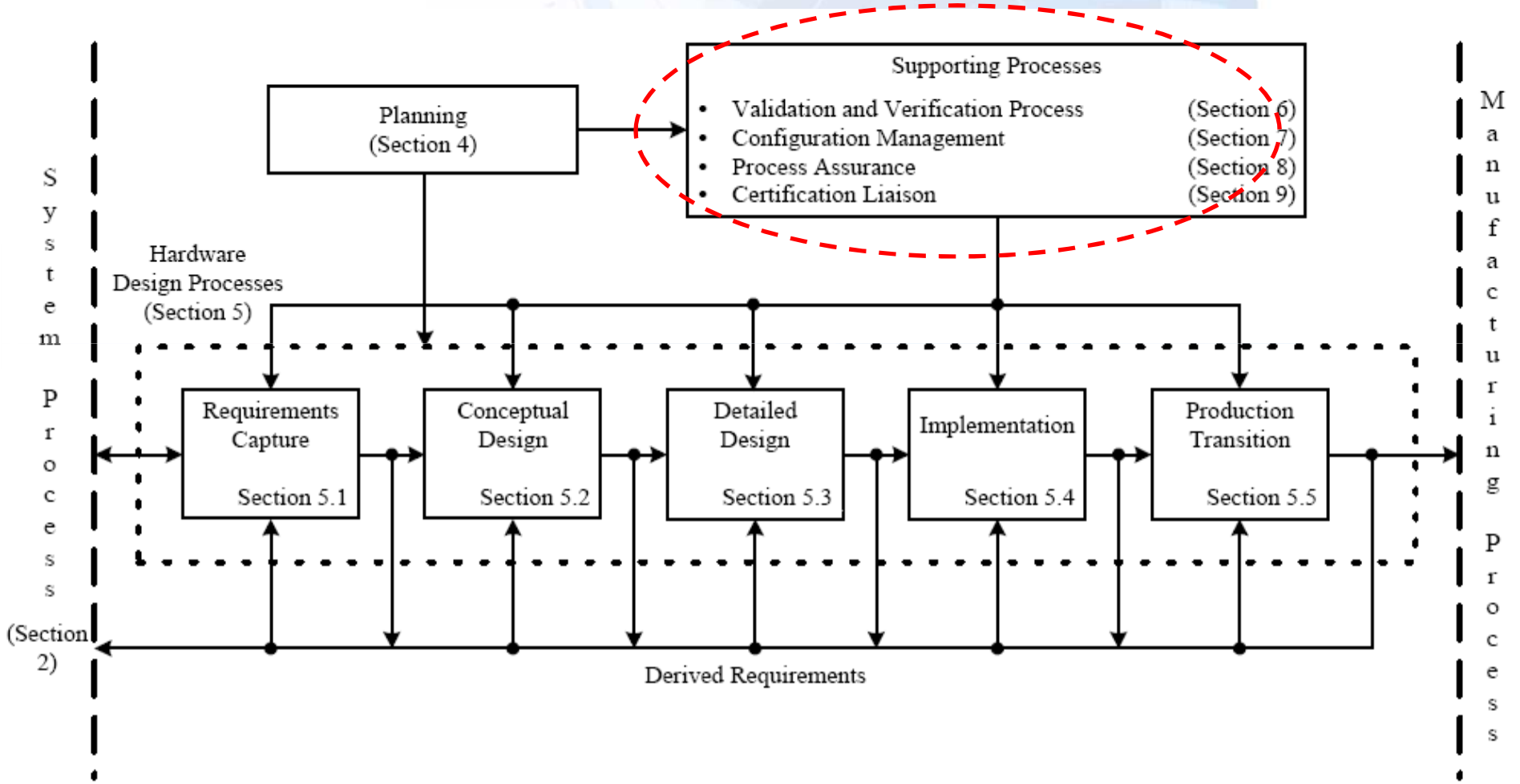
Manufacturing data, test facilities and general resources should be examined to ensure availability and suitability for production

1. Each hardware item produced should be tested by ATP (Acceptance Test Procedure)
2. The ATP should be consistent with safety requirement through FMEA
3. A coverage matrix against requirements should be produced with ATP
4. Each design/in service modification should be evaluated for ATP impact.

Objectives of the training

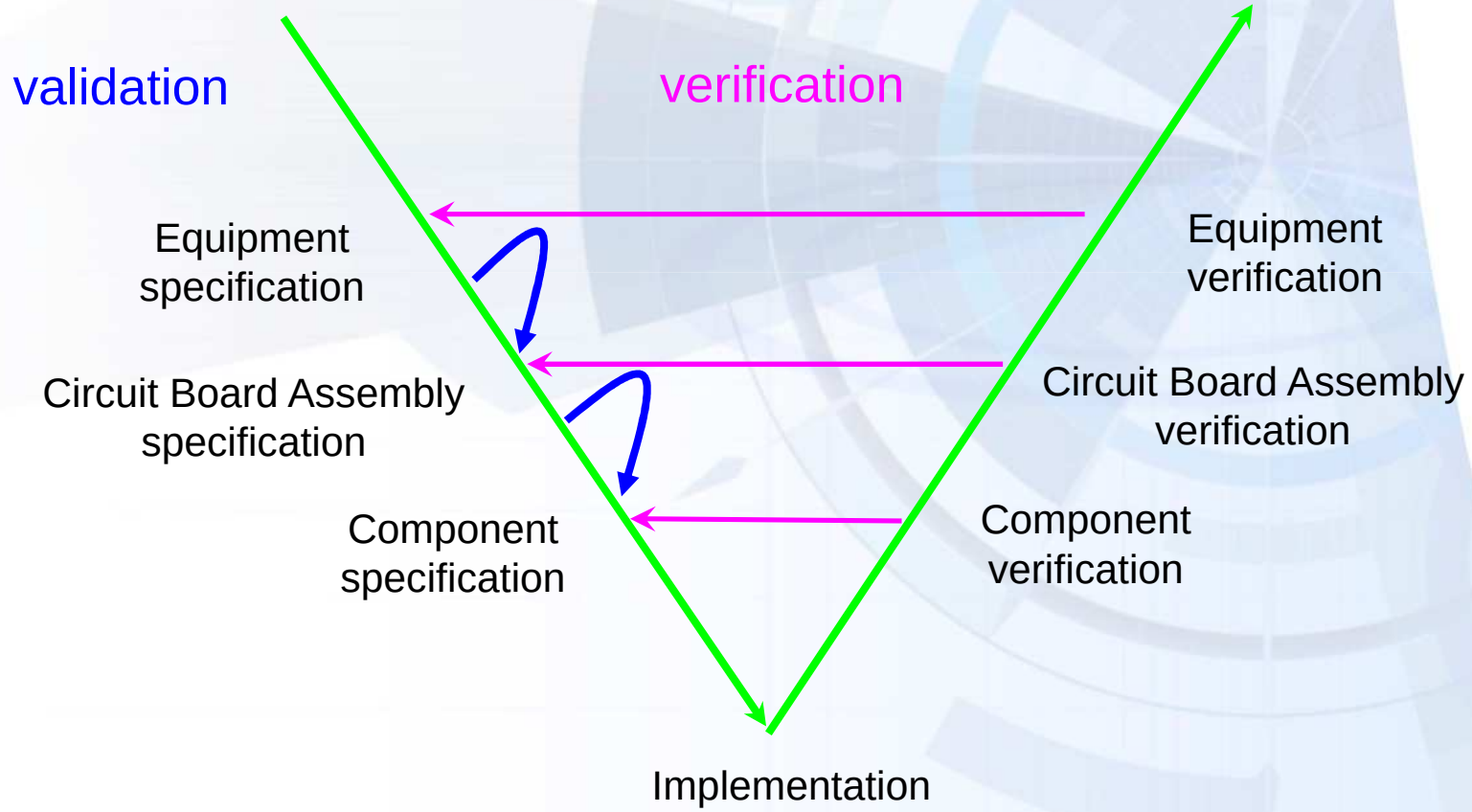
- Presentation of the DO-254
- Complexity considerations
- Hardware planning process
- Hardware design processes
- Supporting processes

Supporting process (1/26)



Validation and Verification

Supporting process
Validation and Verification processes



Supporting process
Validation & Verification
Coverage Analysis

FUNCTIONAL
Coverage

- Every Requirement tested or verified by analysis

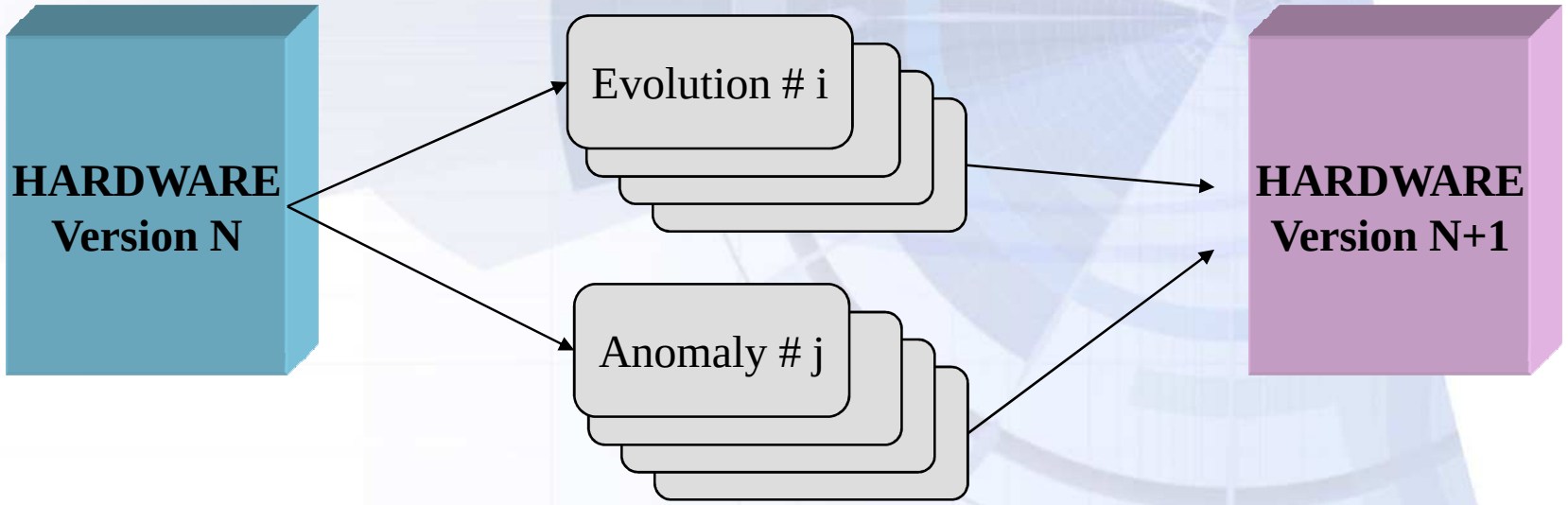
CODE (structural)
Coverage

- Every “structure” of the code is exercised by testing

Configuration management

Supporting process
Hardware Configuration Management
Changes Management

Changes management allows TRACEABILITY between two Configurations



Process Assurance

Supporting process
Process Assurance

Quality Assurance

Objectives

Assurance that the Plans are applied as defined and transition criteria are fulfilled

Main Activities

- ***Audits during Development***
- ***Compliance Reviews***

Data Produced

- ***Quality Records***

- ***Independence***
- ***Authority***

Certification liaison

Supporting process
Hardware certification liaison

Certification Liaison

Objectives

- *Establishment of the links with Certification Authorities*
- *Final Approval*

Main Activities

- *Documentation submission*
- *Reviews Preparation*

Data Produced

- *PHAC*
- *HAS*
- *HCID*

- *Early contact*
- *Confidence*