

# A LOGICAL FRAMEWORK FOR THE ISO26262 SAFETY CASE

Andrea PIOVESAN, Fiat Research Centre

*Powertrain Research & Technologies*

Dr. Edward GRIFFOR, Chrysler Group LLC

*Walter P. Chrysler Technical Fellow*

Software Certification Consortium Meeting 13  
at HCSS 2014 – May 4-6, 2014



# Agenda

- Motivation
- What is a Safety Case?
- Main difficulty in developing an effective safety case
- Proposed Logical Framework for SW Standards
- GSN as a starting point for ISO 26262 Safety Case Framework
- The Framework construction
- Summary, Conclusions, and Future Work

# Motivation: Introduction to Safety Case for complex systems

**Current EE systems are complex**

**ISO 26262 Safety Case is developed to capture an argument and evidence that the system as designed and developed achieves SW Safety Goals**

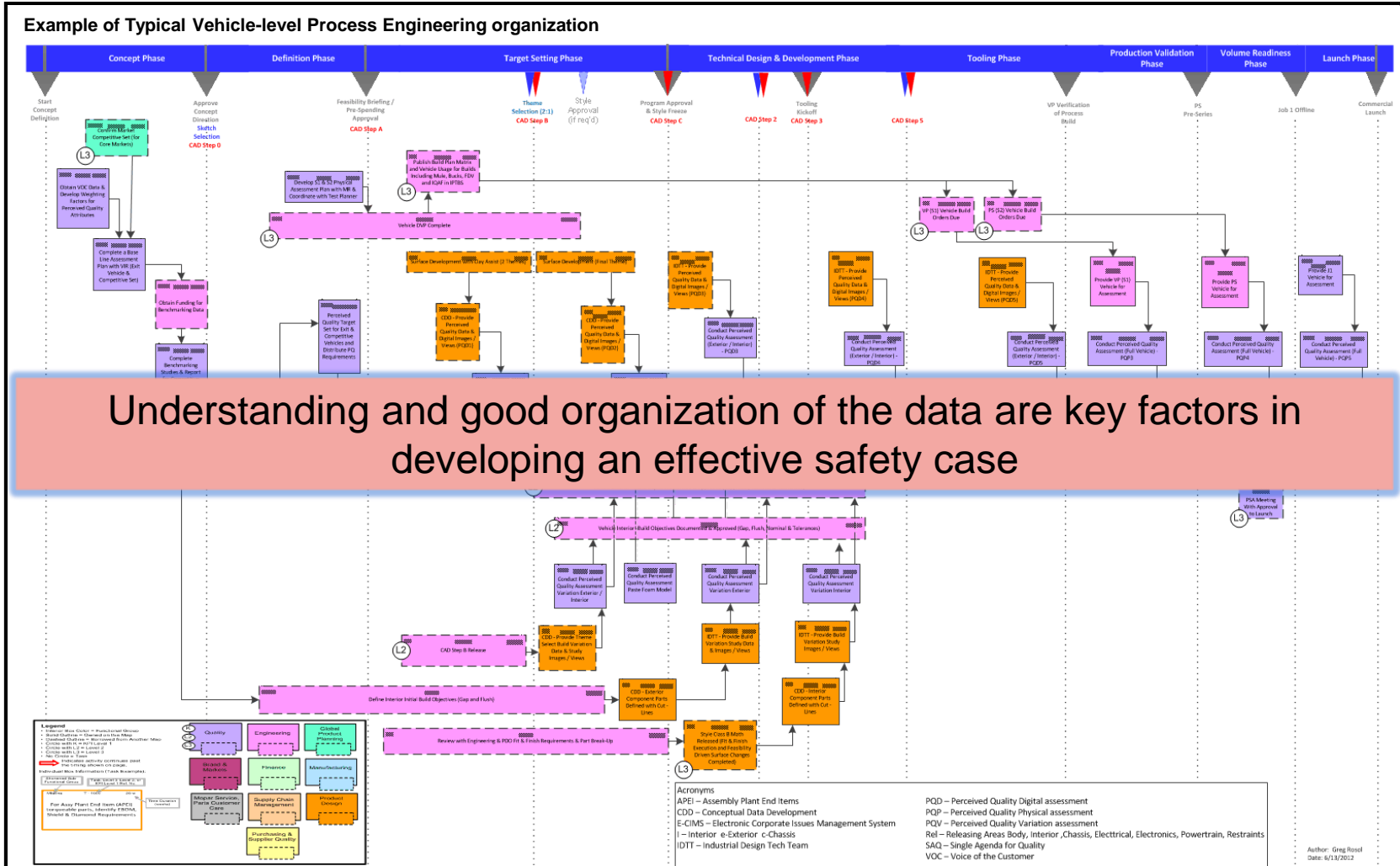
**Developing a traditional text-based safety case is a difficult task**

**Factors that influence safety case development:**

- Increasing number of functions implemented by software
- Wide range of operating conditions and scenarios
- Complicated product development processes including safety lifecycles
- Participation of many people/departments/organizations in the product development and industrialization processes
- Time-to-market constraints

**These factors result in a variety of decisions, activities, processes and documents generated along the product lifecycle**

# Management of Complexity

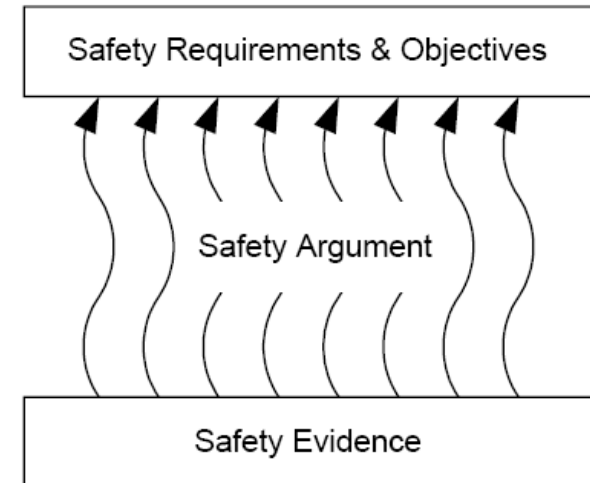


# The Safety Case

*Primary purpose of the Safety Case is to demonstrate and communicate a clear, comprehensive and defensible argument that the “safety properties” of a system are satisfied*

A safety case should consists of three principal elements (see Kelly and Weaver [1]):

- Requirements and objectives,
- Argument *and*
- Evidence.



The Safety Argument establishes and communicates the relationship between evidence and objectives.

# Difficulties in Developing a Safety Case

- ☺ A large variety of definitions, methodologies and technics are available for the Safety Case where:
  - ☹ a commonly observed difficulty is related to the role of the Safety Argument
  
- ☺ Identifying evidence (analyses, test reports, etc.) and objectives (safety goals) are is more clearly addressed while:
  - ☹ a clear statement of the safety argument explaining how the evidence supports the conclusion or provides confidence that the system meets the safety goals is implicitly left to the reader's experience and technical skills

Example: In most organizations a considerable number of safety analyses are performed and available: (FMEA, FTA, etc.), test reports and review reports documenting how specific requirements have been achieved.

However, no rationale is given showing how the combination of evidence demonstrates satisfaction of the safety goals set for the system

# A Logical Framework for Approaching SW Certification Standards (e.g., ISO 26262 Safety Case)

The primary goals of a Logical Framework for the Safety Case are:

- To provide clear criteria for whether evidence and argumentation satisfy the safety properties (safety goals) of the system
- To provide structure for managing and assessing the safety case efficiently and effectively across organizations and throughout a given organization

To achieve these goals, the organization of safety arguments and related evidences is divided into two main categories:

- Product-related arguments that are project dependent
- Process-related arguments that are project independent

# Proposed Logical Framework for the ISO 26262 Safety Case

The Logical Framework for the ISO 26262 Safety Case uses process results and work products, generated during a development process consistent with ISO 26262 where:

- Product evidence shows the system has the required behaviour (satisfies the identified safety goals)
- Process evidence shows the process adopted to develop the product provides the confidence and consistency of the product evidence
- Process arguments are separated from product arguments to facilitate their consistent reuse across a variety of systems




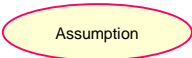



# The “Safety Case Tree”

A graphical notation (derived from the Kelly and Weaver <sup>[1]</sup> Goal Structuring Notation GSN) is our starting point to represent the “tree of steps” in the argument

It gives an intuitive and clear representation of the ‘geometry’ of inference relationships between safety goals and the relevant “proofs” certifying their achievement as well as the induced decomposition of safety properties

## General Purpose of GSN

To show how **goals** (claims about the system)  are broken down into sub-goals, and eventually supported by evidence **(solutions)**  whilst making clear the **strategies**  adopted, the rationale for the approach (assumptions, justifications)  and the **context**  in which goals are stated.

# Assumptions for Safety Case Logical Framework

The goal is to approach a Safety Case using a goal structure with the following assumptions:

- The satisfaction of safety goals, for the Item/System under development, is the primary **goal** of the safety case
- Safety or Certification Standard requirements (e.g. ISO26262 Clauses) and best practices are used to argue/justify (**strategy**) the inference between a goal and its supporting or “sub” goal(s)
- Achievement of safety goals and sub-goals is witnessed by proper **solutions** (evidence) documented by Workproducts (listed in the Safety Plan or gathered from customer’s workproducts)
- **Confirmation measures** are included to argue correctness formally and with respect to contents, adequacy and completeness (e.g. with respect to ISO26262 requirements)
- **Verification reviews** are included to argue correctness, completeness and consistency of workproducts with respect to their technical contents

To each solution is associated one or more work products listed in the Safety Plan or DIA documents

# The GSN Tree construction

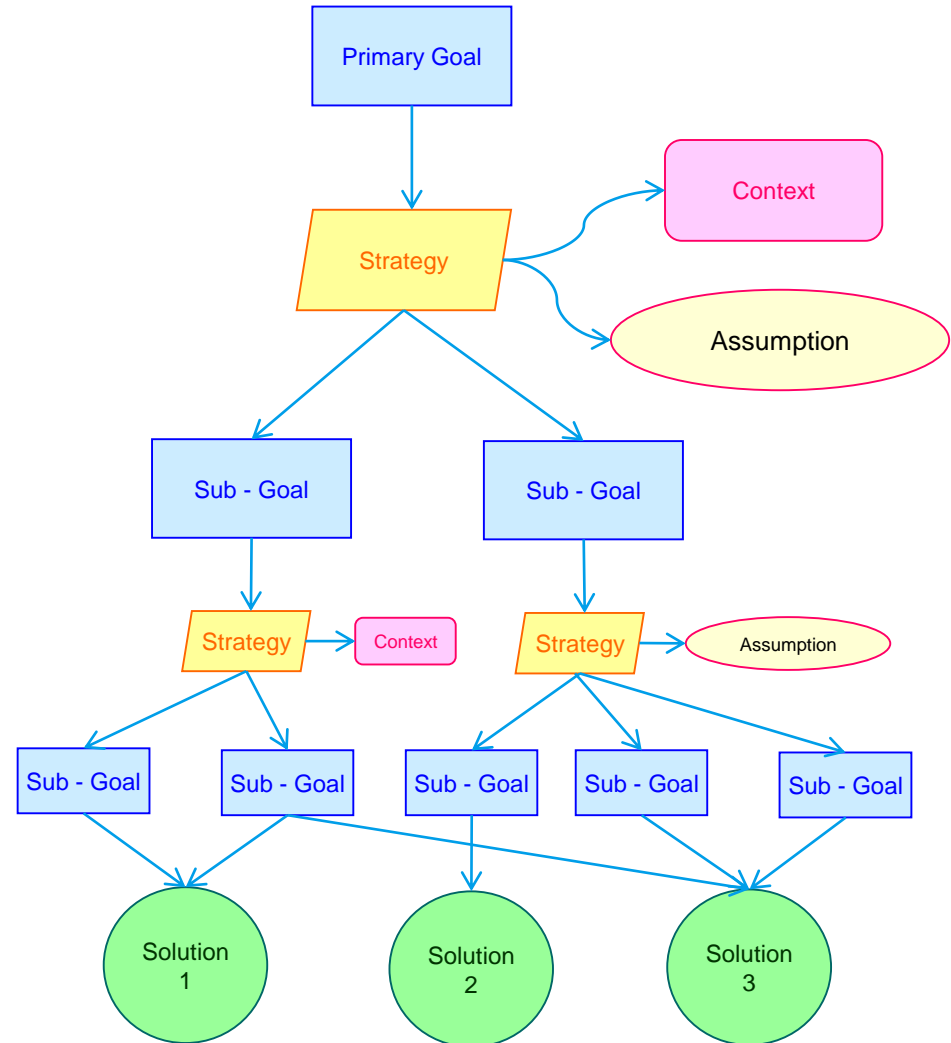
Primary purpose of our Safety Case is to demonstrate the satisfaction of safety goals of the Item under development

Strategies are used to structure the argument that a goal can be inferred from its supporting/sub goal(s) or evidences.

*Strategies typically are product-based or process based arguments/justifications derived from ISO 26262 requirements*

Evidence for goals' or safety properties' having been achieved is obtained through a **progressive simplification** of the safety properties based on the **two-dimensional decomposition** into subsystems, on the one hand, and the logic of ISO2626, on the other, until one reaches a set of "elementary" safety properties whose truth can be directly derived from the set of *Solutions*

*Solutions* are both product and process evidence gathered as parts of ISO26262 consistent workproducts



# Logical Framework of a Safety Case

Primary purpose of our Safety Case Logic is to demonstrate/derive the satisfaction of safety goals of the Item under development. The 'ontology', as noted earlier, consists of:

- Requirements and objectives,
- Argument/Reasoning
- Evidence

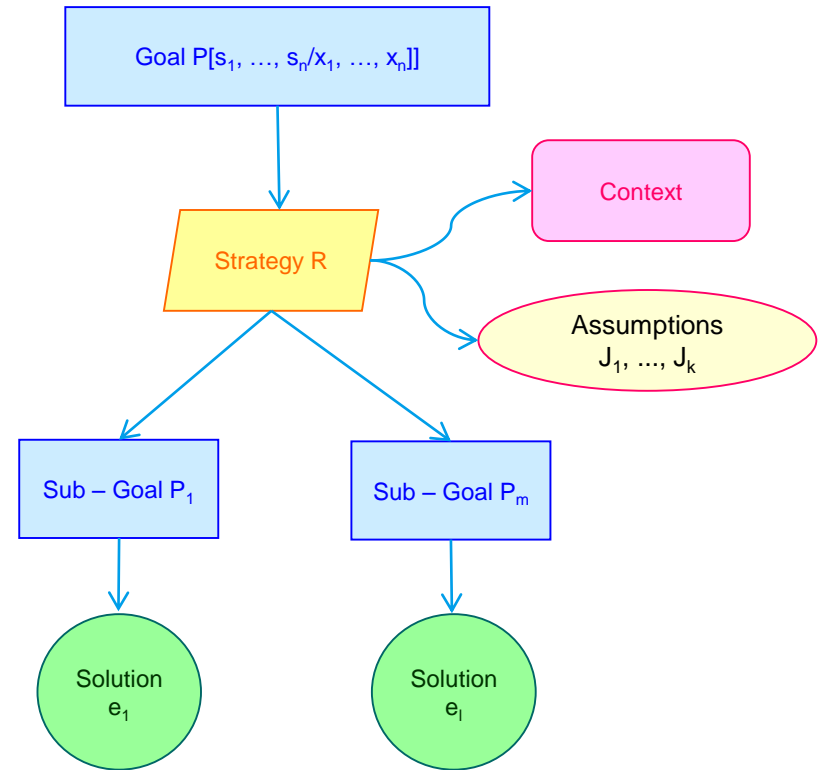
A logical framework is defined by an appropriate syntax/language for the analysis of a system  $S$ , where:

- Goals/Objectives = Propositions (“safety properties”)
- Evidence = constructions/proofs of propositions
- Argument = Rules for deriving statements of the form “ $e$  is a proof of property  $P$ ” that reflect conformance with the Standard

Statements in the Argument have the form:

$$J_1, \dots, J_k \mid - e \in P\left[\frac{s_1}{x_1}, \dots, \frac{s_n}{x_n}\right]$$

Which is read “from statements  $J_1, \dots, J_k$  we can derive the fact that evidence  $e$  is evidence for the fact that  $P$  is true of systems  $s_1, \dots, s_n$ ”, where  $x_1, \dots, x_n$  are the system variables of  $P$ .



This inference would have the form:

$$(\text{Appl. of Rule R}) J_1, \dots, J_k \mid - e \in P\left[\frac{s_1}{x_1}, \dots, \frac{s_n}{x_n}\right]$$

Where  $e = \langle e_1, \dots, e_n \rangle$ , an encoding of solutions  $e_1, \dots, e_i$  and  $R$  is the rule of the framework for the decomposition of *safety property*  $P$  into  $P_1$  thru  $P_m$ .

# Summary, Conclusions and Future Work

This presentation provided an overview of the approach to Software Systems development that progressively creates a safety case that:

- Requires minimal extra-effort in terms of time and resources;
- Makes easy re-use and adaptation for similar projects (changes, carry over);
- Is more easily understandable (GSN is used to give an intuitive and clear representation of the inference involved and a Logical Framework is used for working with 'safety properties')
- Is highly flexible for application to different software certification standards, project applications, as well as business models and organizations.

This approach uses process results and work products that are generated during a development process consistent with a Software Certification Standard (such as ISO 26262) in order to successfully demonstrate that the inference involved, between the safety goals defined for that product and the supporting evidence for the adequacy of the analysis, are sound and are consistent with that Standard.

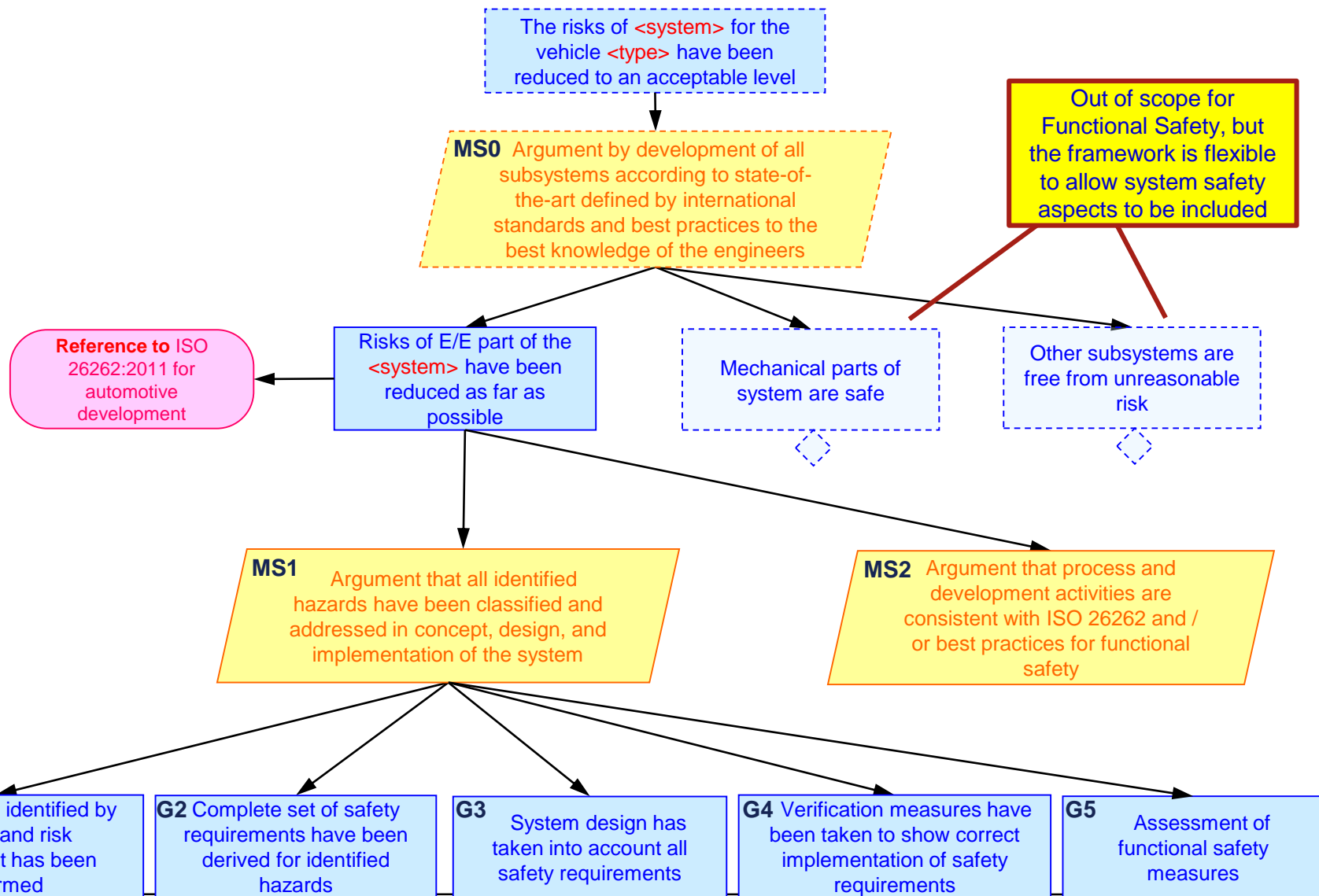
It is our assessment that a theoretic and simple extension of this logical framework for the safety case of a Software Certification Standard to any specific safety standard and technological domain will be possible (e.g. industry, aerospace, etc.). In our forthcoming paper *The Logical Framework of the Safety Case* we include details of the application of this method to *Advanced Driver Assistance Systems*.

# References

- [1] Kelly T. and Weaver R., *A systematic approach to safety case management*, in CAE Methods for vehicle crash worthiness and occupant safety, and safety critical system, 2004 World Congress Special Publication SP-1879, Society of Automated Engineers, 2004
- [2] Raghad Dardar, Barbara Gallina, Andreas Johnsen, Kristina Lundqvist IDT, MRTC, Mälardalen University, Västerås, Sweden *Industrial Experiences of Building a Safety Case in Compliance with ISO 26262*

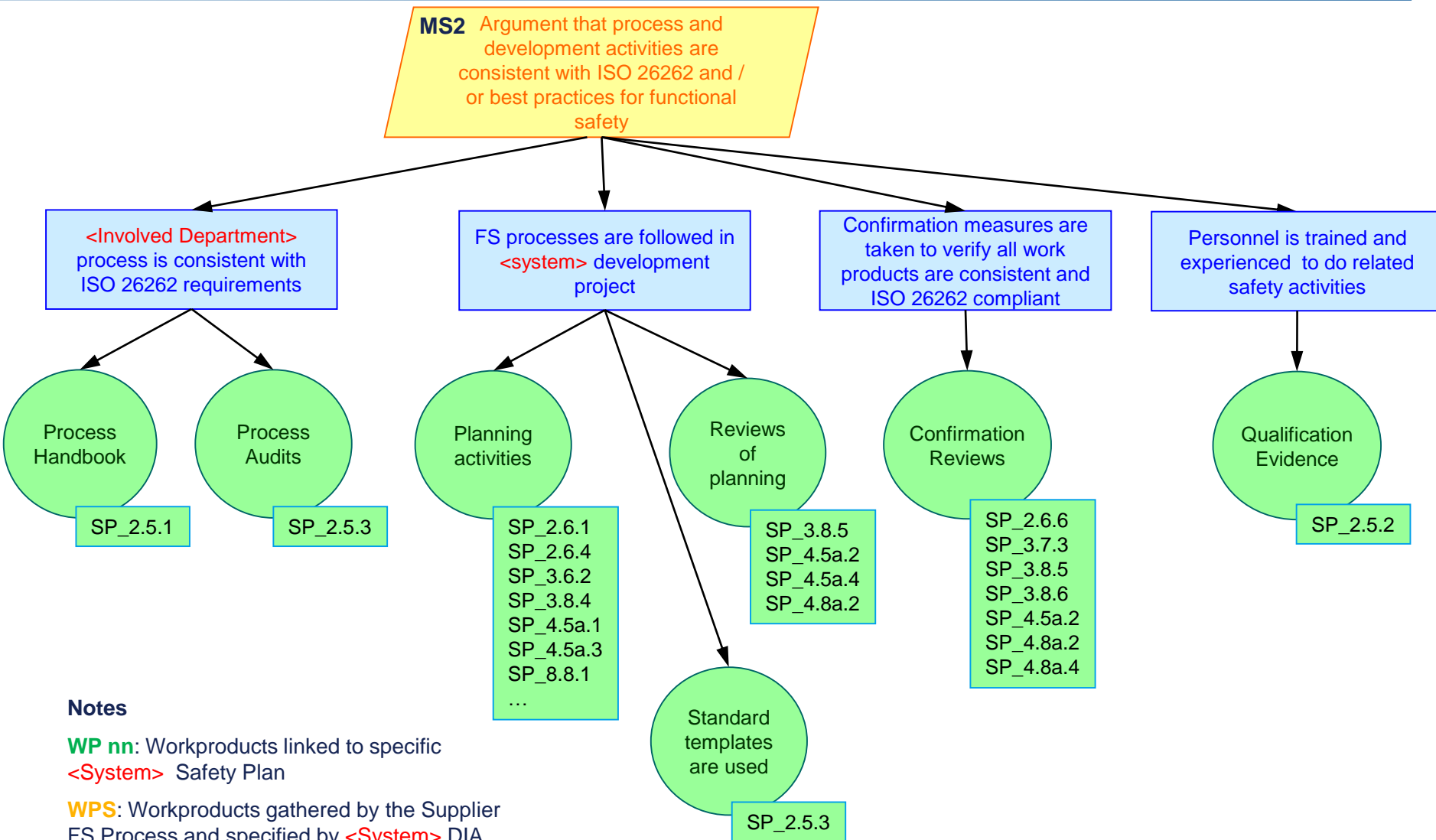
# Back-Up Slides – Detailed Tree Representation of the Safety Case

# The Root of the Safety Case “Tree”

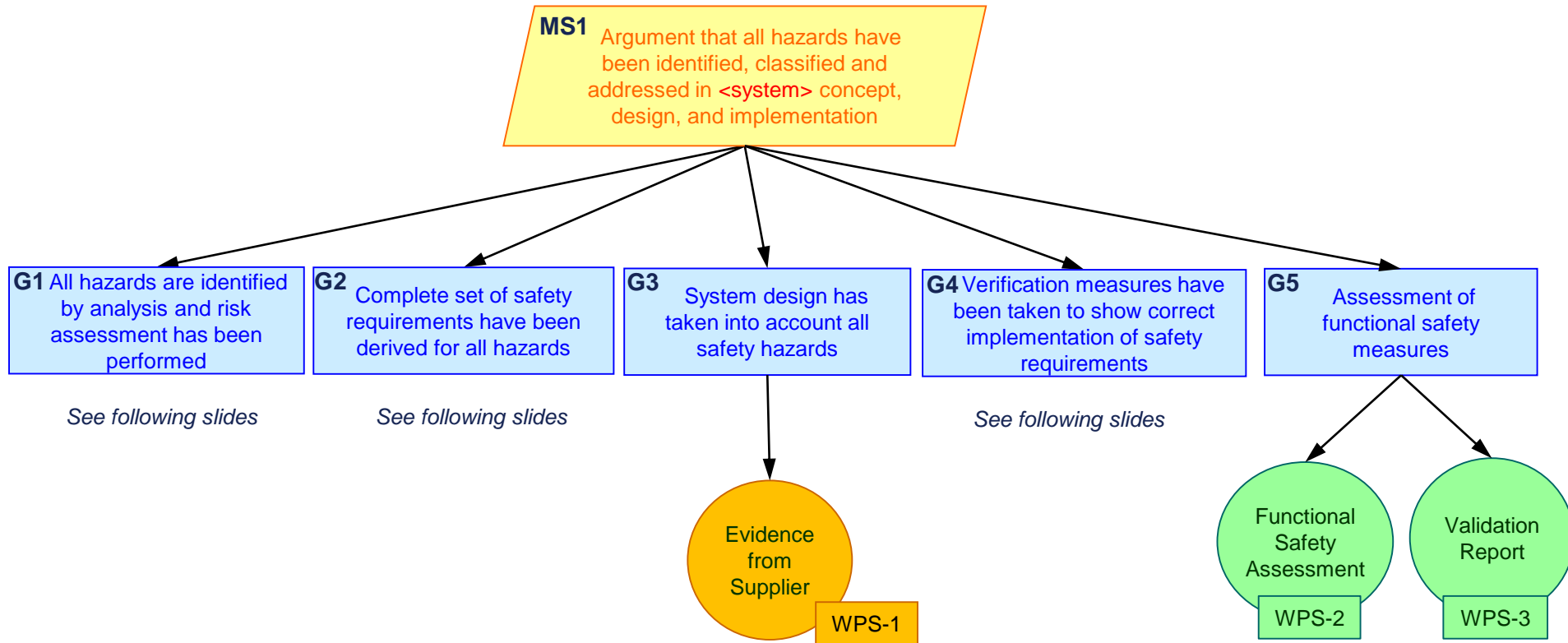




# Logic of the Safety Case: Process argumentation



# Logic of the Safety Case: Hazard argumentation



## Notes

**WP nn:** Workproducts linked to specific  
<System> Safety Plan

**WPS:** Workproducts gathered by the Supplier  
FS Process and specified by <System> DIA

# Logic of the Safety Case: Hazard identification

**G1** All hazards have been identified by analysis and risk assessment has been performed

## Notes

**WP nn:** Workproducts linked to specific  
<System> Safety Plan

**SC1** Argument completeness of hazards by analysis of all malfunctions in combination with relevant driving situations

Hazard analysis and risk assessment of ISO 26262-3 is sufficient for identification of hazards

Functions and typical operation situations in scope have been defined

Assumptions on behavior and interaction with other vehicle systems was known

Malfunctions have been determined for item functionality

Relevant operational situations have been considered in the analysis

Hazards have been derived consistently from hazardous events

Item Definition  
WP1

HAZOP is sufficient for finding malfunctions

Argument by HAZOP method

Argument by experience

Malfunctions are completely identified

List of standard operational situations is maintained

Systematic combination to hazardous events in analysis

List of standard operational situations  
WP2

HARA Worksheet  
WP3

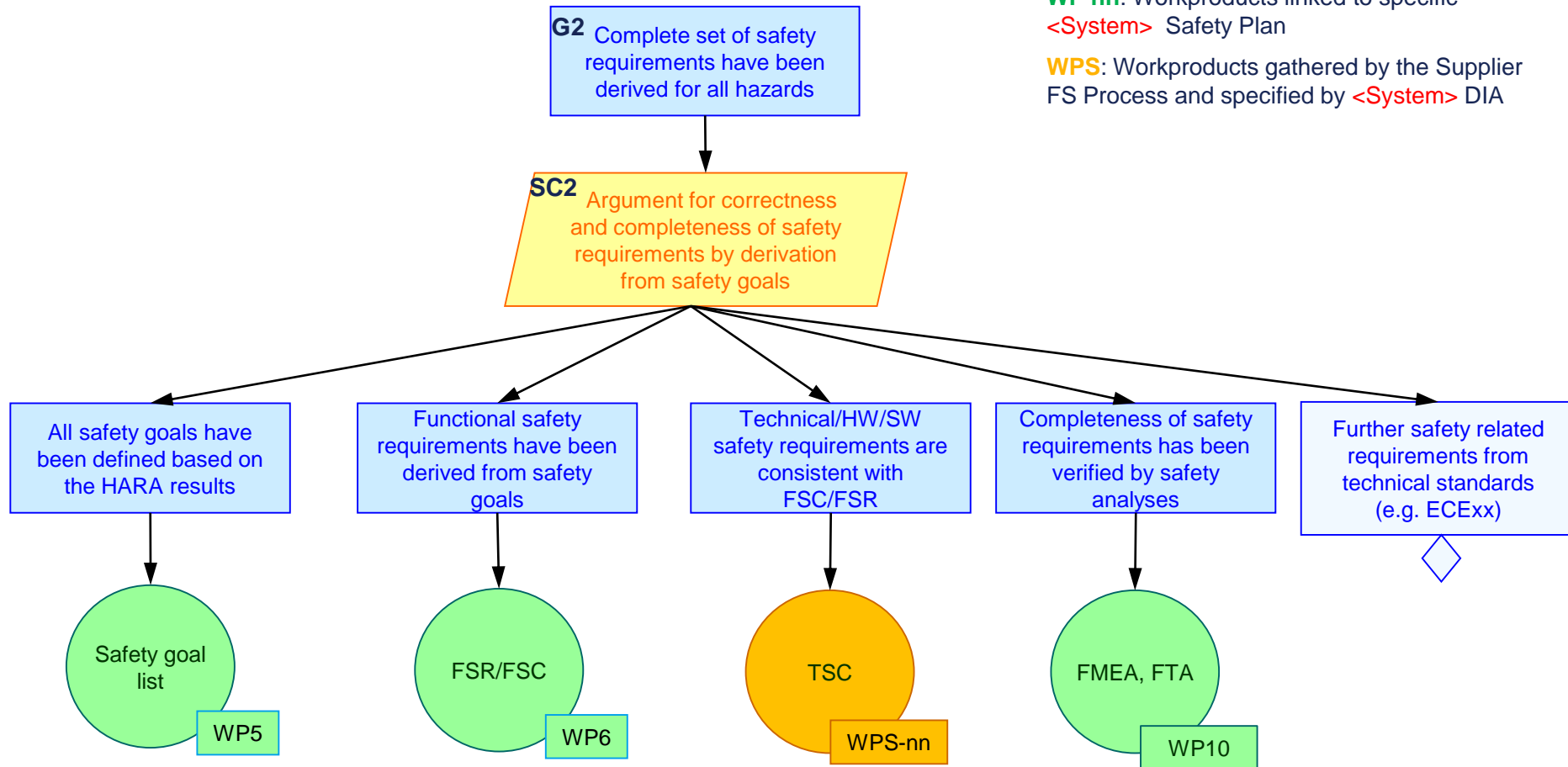
Evidence on S/E/C  
WP4

# Logic of the Safety Case: Requirement elicitation

## Notes

**WP nn:** Workproducts linked to specific  
<System> Safety Plan

**WPS:** Workproducts gathered by the Supplier  
FS Process and specified by <System> DIA

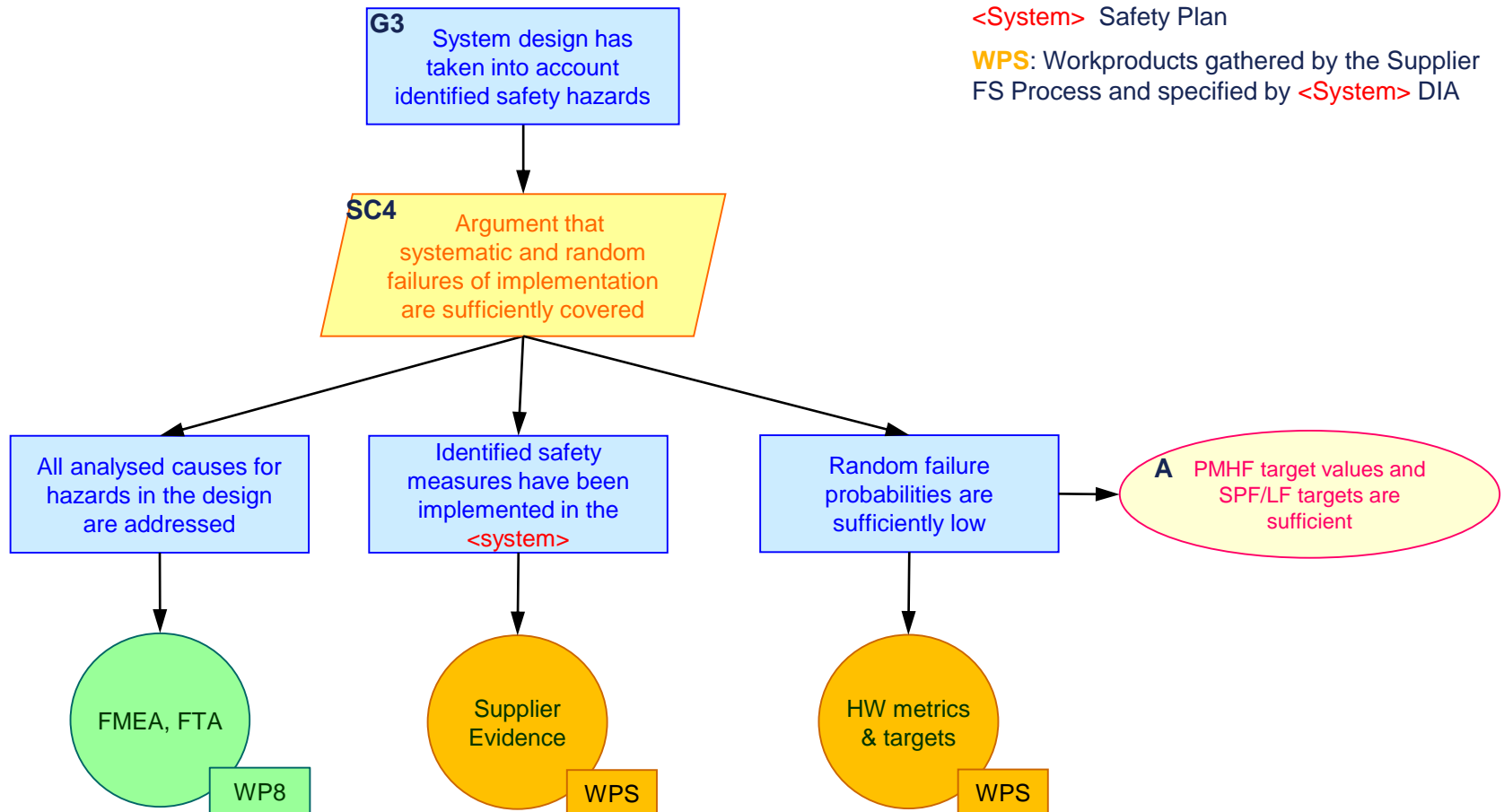


# Logic of the Safety Case: System Design

## Notes

**WP nn:** Workproducts linked to specific  
<System> Safety Plan

**WPS:** Workproducts gathered by the Supplier  
FS Process and specified by <System> DIA



# Logic of the Safety Case: Verification measures

