INCOSE

WESTERN STATES REGIONAL CONFERENCE

# Safety Requirements Management Process in Aerospace System Design

INCOSE WSRC · 2025

**AUTHORS**

Manji Maheve

**PUBLICATION DATE**

July, 2026

**SUBMISSION #**

3

INCOSE PUBLICATIONS

# Paper Title: Safety Requirements Management Process in Aero-space System Design

**Manju Maheve**
**Collins Aerospace**
**27300 SW Parkway Ave,**
**Wilsonville, OR, 97068 USA**
**Manju.Maheve@gmail.com**



## Abstract

System requirements management is one of the most important aspects of any robust and complex system design. The increased complexity of a highly integrated system with safety attributes at the core of an aerospace system design warrants for a very detailed requirements management process that caters to the functional and safety requirements originating at aircraft level. Along with this it also needs to take care of the specific derived functional and derived safety requirements originating at the different phases of the system development life cycle that includes system architecture review and safety analysis processes. The recommended practices of ARP4754B and ARP4761A provide a framework for system development life cycle and safety assessment process but does not detail any standard approach for requirements management, their attributes handling and particular activities for safety requirements which makes the requirement's process susceptible to variations and individual interpretations leading to rework and increased cost of development as observed in the Aerospace Industry.

This paper presents an approach for effective system requirements management and specific activities required for handling safety requirements as recommended per ARP4754B and ARP4761A. The paper has summarized the safety assessment process flow and safety requirements flow down to hardware and software requirements. It has also identified common issues and pitfalls in safety requirements management. The paper proposes best practices to leverage the process maturity and reduce rework and variations. At the end the paper provides scope for further analysis and area of work using emerging methods.

## Keywords

ARP, SR, SDLC.

# Introduction

Aircraft System development is an iterative and phased engineering activity that starts with concept development as shown in Figure 1 – Aircraft System Development Phases.

The development of system starts with the concept phase where the aircraft, its initial configuration and performance are defined for preliminary development. This initial information is leveraged for identifying the systems functions and defining associated systems requirement. The initial information defined at concept phase also helps define the system boundaries and system level interface requirements. The system requirements are allocated to lower-level items (Hardware/Software) as defined in system development process.

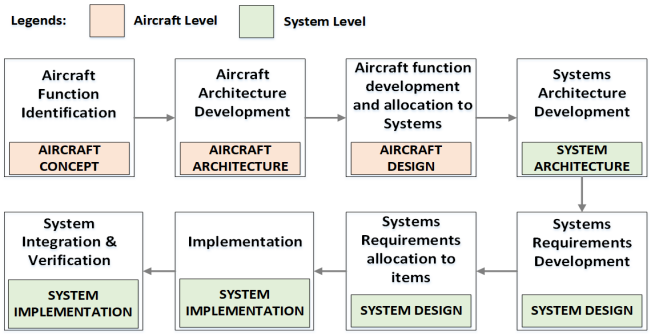


**Figure 1. Aircraft System Development Phases**

The system development process and system design assurance are highly regulated for safety aspects in Aerospace. The certification authorities usually evaluate the system development processes using recommended practices defined in ARP4754B and ARP4761A for gaining confidence on process maturity and provide certification approvals.

ARP4754B defines a system design framework. It captures the high-level processes objectives, outputs, and configuration control categories in Appendix Table A-1 of the document to ensure compliance. It identifies the interaction of System Design Framework with software and electronic hardware development life cycle processes during system design and implementation as discussed below.

## ARP4754B System Design Framework

ARP4754B emphasizes the need to integrate Safety assessment process and Systems development process throughout the equipment and system development life cycle. The system level requirements and interfaces are decomposed to arrive at the functional and physical architecture. Safety assessment process starts with the architecture design and system functional requirements as the initial input. The architecture and system requirements are used for the detailed design and allocated for implementation of items at hardware or software levels. Other than the safety assessment process, ARP4754B outlines the relationship between the hardware and software development documents, which provide guidelines for electronic hardware and software life-cycle processes. Figure 1 in ARP4754B framework depicts system design framework in integration with the guidelines as defined in DO-254, DO178 and DO-297.

The safety assessment process seeks input from system development process for safety assessment and on completion of activities as per ARP4761A provides the output from safety process in terms of additional functions, failures and safety information back to the system development process as shown in Figure 1 in ARP4754B Framework. The different types of safety assessment process and their impact on the requirements management process for handling the functions, failures and safety information are discussed in detail in below section.

This whole process of flow down of aircraft function allocation to systems, system requirements, architecture development, detailed design and implementation are covered by development assurance process to mitigate the risk of development errors, undesirable and unintended effects in developing a complex system.

## Safety Assessment Process

The Aircraft System Development Phases right from the concept are continuously covered by the safety assessment which not only assesses the aircraft functions and systems performing these functions to identify all failure conditions and their associated hazards. It also evaluates any additional complexity and interdependencies arising out of integration of systems and functions and their further impact on the aircraft safety and hazards.

These assessment leads to origination of additional safety and derived safety requirements. A

complete flow of the safety process is as shown in Figure 2 - Safety Assessment Process.

The general coverage of different types of safety assessment activities to be performed at different phases of the system development life cycle are presented below.

- Functional Hazard Assessment (FHA): This assessment is conducted at Aircraft and System level where failure conditions are identified and severity of failure on the aircraft safety is evaluated for categorization – Catastrophic, Hazardous, Major, Minor and No Effect
- Preliminary Aircraft Safety Assessment / Preliminary System Safety Assessment (PASA / PSSA): This activity requires to inspect architecture and implementation to ensure safety requirements for all items identified for failure conditions during Fault Tree Analysis (FTA)/FHA are covered. The missing requirements are added as additional safety requirements and PASA/PASA are updated iteratively multiple times till the complete coverage of failure conditions are achieved. PASA/PSSA is method to evaluate system architecture and derive safety requirements.
- Aircraft Safety Assessment / System Safety Assessment (ASA / SSA): It is the verification activity to ensure the implementation of the system meets its defined safety objectives and requirements as per FHA and PASA/PSSA both qualitatively and quantitatively. The shortcomings identified during ASA / SSA could lead to additional iterations of PASA/PSSA, addition of new safety requirements and incorporation of these new safety requirements in the system architecture and implementation along with documented evidence for the test, demonstration and inspection.
- Common Cause Analysis (CCA): CCA requires to evaluate the functions, systems, or items for their independency and to identify the individual failure modes or external events that could lead to failure of other functions, system or items defying the claims of independency. It requires the following different type of analysis
    - Zonal Safety Analysis (ZSA): It requires to analyze systems which are installed in the same zone and do not impact each other's safety due to physical proximity and mutual influence. It should take care of safety requirements specific to installation, interference between systems and maintenance errors and their effect on aircraft.
    - Particular Risk Analysis (PRA): This requires evaluating very specific risks which are outside the influence of system concerned and requires to identify risks which could cause cascading or simultaneous failures (Example: bird strike, lightning, fuel leakage, tire burst). The particular risks are more circumstance dependent and technology independent.
    - Common Mode Analysis (CMA): The objective of this analysis is to ensure that a common cause or error should not lead to outage of two or more items or systems. This activity proves the independence of functions and their respective monitors by verifying that failure events identified in the ASA/SSA are independent in their implementation.

The system safety assessment yield safety requirements and adds additional derived safety requirements which could lead to design changes and hence is an iterative process of management of safety risks to either eliminate, control or accept [4] within the purview of operational effectiveness of the aircraft and its system.

The complexity of system varies based on aircraft level failure conditions and degree of item level integration for different aircrafts. The safety requirements management process witnesses a lot of variation in integration of the processes to overall system development life cycle. This leads to increased cost both in engineering and production, resulting in a need for standard and optimized process.

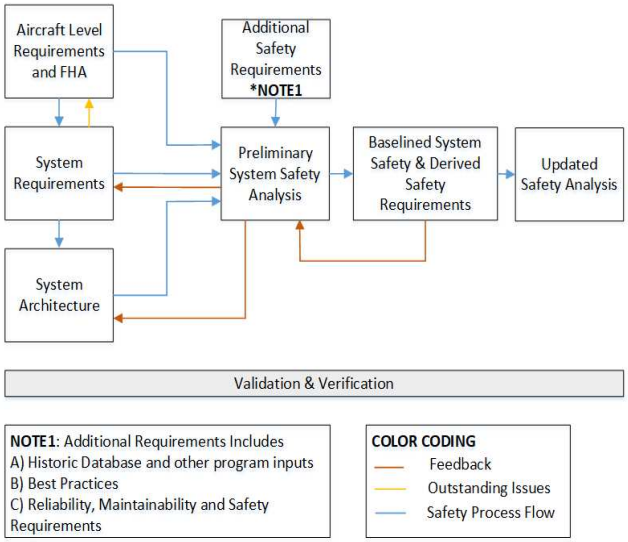


**Figure 2. Safety Assessment Process**

## Safety Requirements Capture and Management

Requirements capture and management process defines the planning, implementation, and evaluation of system requirements and derived requirements throughout the product development process. The linkage and flow of this process is shown in Figure 3 – Safety Requirements Flow-down. The safety related requirements are in general related to customer and regulatory requirements and are flown down to hardware and software design. The safety requirement management process needs to ensure tracing and closure of the requirements to the rigor needed per the system criticality and certification criteria. Safety management team is responsible for managing the requirement tags and tracing to the requirement source for all the safety requirements. The safety requirements review starts in parallel with the preliminary system safety assessment by the safety engineering group. The first step is the validation of customer safety requirements in view of aircraft level safety requirements and regulatory requirements. Safety requirements for historic programs and best practices & lessons learnt are also the considerations while validating the system safety requirements.

Preliminary system safety assessment brings out additional scope for identifying and generating safety requirements. All of these added requirements are then classified into system/safety requirements and derived safety requirements. Some of these requirements need to be added at the system level and should be communicated back to the aircraft level.

Once the Safety Requirements (SR) are identified from the Safety Assessment process, these SRs should be added in same database with other system requirements. They need to follow the same life cycle but require additional safety attributes and processes for their definition and management.

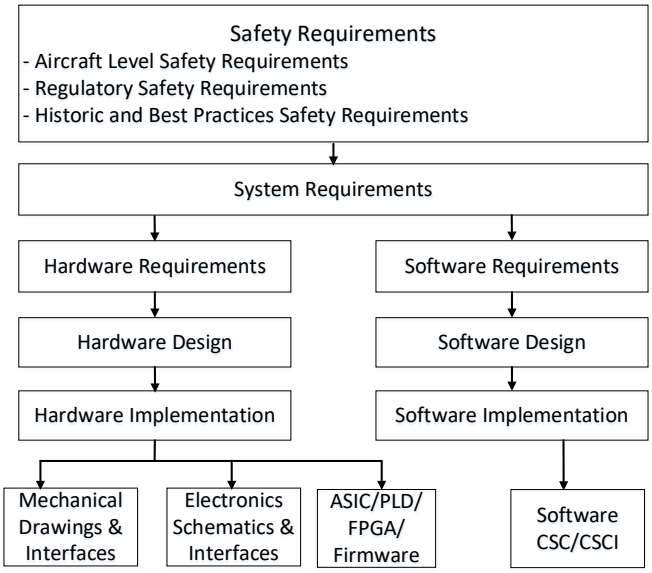


**Figure 3. Safety Requirements Flow-down**

# Safety Requirements

The terms Derived Requirements and Safety Requirement (SR) are used casually but they are slightly different. In the System, Hardware (HW) and Software (SW) domains, “derived requirement” refers to a new requirement which is not directly derived from an existing higher-level requirement.

Derived requirements usually contain implementation details needed to make the product

function correctly. In a complex LRU, there may be thousands of derived requirements. They are created to address characteristics which are not part of the higher-level requirements. Therefore, in Hardware/Software, there is NO linkage between a derived requirement and any higher-level existing requirements.

A Safety Requirement is a real requirement which is derived from the Safety assessment (PSSA). It describes an attribute of the implementation necessary to meet the safety probability requirements or to develop an independence requirement. Safety requirements may not directly be traceable to higher level requirements, though derived requirements can influence higher level requirements.

ARP4754B recommends that derived requirements "be captured and treated in a manner consistent with other requirements applicable at that development phase". It is good practice to treat safety related functional requirements in a similar way, since they are subject to the same obligations as other requirements with respect to traceability.

The PSSA brings out the need for implementing the function independence, need for specific Functional Development Assurance Level (FDAL) and requirements to implement the sufficient hardware or software monitors for fault detection and correction. Thus, the safety requirements can be classified per the safety objective.

## Types of Safety Requirements

Safety Requirements are based primarily on AND gates and some Basic Events which appear in Fault Trees (FT) in the PSSA. Safety Requirements are written to address a System, Hardware or Software characteristic of the design which is modeled in the Fault Tree Analyses (FTA) or otherwise required by the PSSA. The Safety Requirements should be clearly stated and have sufficient context to prevent misinterpretation.

Safety Requirements for redundancy and monitors are commonly derived from AND gates in the FTA. Additional Safety Requirements are for tests, fault response, annunciation, etc. which originates directly from PSSA.

Following are the major types of SRs

1. Safety Requirement for Functional/Item Development Assurance Level (FDAL/IDAL): All FDAL/IDAL allocations requirements shall be Safety Requirements. It is strongly recommended that FDAL allocations be documented in the PSSA using Functional Failure Sets (FFS) whenever a FDAL lower than the top-level FDAL for the identified hazard is proposed.
2. Safety Requirement for Architectural Independence: There shall be Safety Requirements for architectural independence. Safety Requirements for independence shall list the Hardware which must be operationally independent.
3. There shall be a Safety Requirement for each monitor proposed in the PSSA. State the need for the monitor in the Safety Requirement comment. This should be a descriptive text that adds context. Also state how the monitor is expected to operate. These comments are valuable to the designers who must implement the monitor.

### Early safety and late safety requirements

When a program is being planned, and the system definition is in draft; it is difficult to perform detailed safety analysis. Often system descriptions are broad and early PSSA FTAs are based on informal information rather than released design documents. At this early stage of development, Safety Requirements which state functionality (not hardware) are appropriate. The creation of preliminary Safety Requirements at this stage, may not necessarily result in each specific early phase Safety Requirement being incorporated into the final PSSA (late-stage SRs). Safety Requirements created to support early phases of development may state required safety characteristics in broad terms.

Early phase Safety Requirements may be captured in an ad hoc table and distributed to designers. These early phase Safety Requirements may or may not become part of a Draft PSSA or PSSA, depending on the final design. In the later stages of development, the design matures, and more details of the design are documented by Sys, HW and SW and signal flow diagrams are available as references for defined system functionality. This allows the PSSA or Draft SSA FTAs, and therefore Safety Requirements, to reflect the design and implementation more closely. Safety Requirements created during the late phases should be directly

linked to specific FTAs. Prior (early phase) Safety Requirements may be superseded, enhanced or deleted, depending on the current design.

## Observation from current practices

**Involvement of Safety Engineering late in the design phase:** Most of the time it is seen that the Safety and Reliability team is involved late in the design. Once the design is ready which means System architecture, BOM/Schematic are ready, the Safety and reliability team is asked to perform the analysis. This late involvement might result into some design gaps identified from the Safety and Reliability analysis that resulted into System/product being non-compliant to safety requirement.

Recognizing the iterative nature of new product design process and the ARP4754B Integral Processes, the safety engineer is directed to participate in an interactive and mutually advantageous relationship with the design engineers. This results in proposed new designs being quickly evaluated for safety impact.

**Incomplete coverage to Safety Requirements:** It generally consists of multiple elements (FTA AND gates) which resulted into number of Safety Requirements created. A large number of Safety Requirements pointing to same function could result into traceability issues while flow down to design (hardware/software).

**Incorrect Safety Requirements:** It is very important to create a valid Safety Requirements. It is necessary to understand that both ARP4754B and ARP4761A state that probability requirements be included in the formal requirements. Considering that, we should not write a Safety Requirement which state a required probability or specify Single Event Effect (SEE) rate as SEE failure rate is based on device size, exposure time etc.

Also, there were instances where a Safety Requirement becomes redundant to existing system requirement which makes Safety Requirement unnecessary. This situation arises when the knowledge or the study of existing system level requirement is incomplete.

**Immature process:** Sometimes the design does not comply with safety requirements and even system development plan/processes fails to provide clarity in handling such cases.

# Proposed Enhancements and Best Practices

In order to address the gaps identified in the Safety Requirements Management, a need for a robust & mature process by implementing some of the best practices is proposed. Some of the aspects are linked to a defined and matured requirements management process for Safety Requirements. Implementing a robust safety management process along the development life cycle is another catalyst that leads to timely integration of safety aspects with the product development phases.

## Safety Requirements Classification and Management

Considering the number of Safety Requirements in complex designs, it is important to manage the safety requirements to establish the linkages and traceability to aid the requirements verification and compliances. Some of best practices to define and manage the safety requirements by defining attributes are described as follows.

### Safety Requirements Attributes

1. **Safety Tags:** The use of Safety Tags in the requirements is the means of ensuring Safety Requirements, and their decomposed elements, are guarded from unintentional modification. The domain requirements process needs to include a method on the use of Safety Tags.
   - Safety tags should be implemented at all levels
   - Safety tags should be set = True for all SRs and subsequent child requirements. This is best performed at the time they are created.
   - A rule must exist which prohibits changes to the SRs or child requirements without the prior consent of the cognizant safety engineer.

   This is the method of ensuring that SRs are implemented correctly and not modified without safety participation.

2. **Traceability:** It is essential that the source of the Safety Requirement be documented. When

questions arise, it is the link between the SRs (as captured in the formal requirements) and the PSSA FTA Gate which allows the safety engineer to understand the rationale for the safety requirements. All SRs shall include a link (reference) to the FT and Gate from which the requirement is derived. There should be a rule in the formal requirements which states that each safety requirement copied into the formal requirements shall include the source information (released PSSA document number and SR number) in the Rationale or Comment field. In the case where multiple AND gates result in the same requirement, a change requested by the HW/SW design engineer needs to trace back to ALL of the AND gates to prevent an incomplete interpretation of a change request.

3. **Content/Wording:** Safety Requirements shall be stated as a requirement containing the word "shall". SRs shall use standard sentence structures where applicable.
4. **Rationale:** This is the justification for the safety requirement.
5. **Validation:** ARP4754B brings a new formality to requirement validation. In the safety domain, this means Safety Requirements. Depending on the FDAL level of the function, the validation may have to be done independently. In the Safety requirements tables, for each requirement, there should be a detailed reference to the Fault Tree and Gate.
6. **Verification:** Verification of Safety Requirements occurs via standard Sys/HW/SW verification methods, depending on where the SRs flows. The SSA is the analysis method of verifying that the implementation is adequate to meet the safety requirements, but the SSA does not verify the safety requirement implementation.
7. **Tools:** The approach for marking and tracing safety critical requirements is dependent on the tool used to compose the requirement.
    - **DOORS/JAMA:** A separate attribute in DOORS shall be specified as "Safety Requirement". This method will be used to ensure derived safety requirements are clearly identified. This attribute can be used to filter on these requirements for identification, reviews, audits, etc.
    - **Other Application:** Excel based, word document

## Process enhancements

This paper is defining and bringing out the importance of managing the safety requirements and derived safety requirements. It is also important to notice some of the limitation in existing practices. Implementing the requirements management process varies across organizations and across product lines. Process flow and the implementation of the requirements management process are limited to the capabilities of the tool and internal process. Following are few proposals suggested irrespective of the tool here.

***Early and timely engagement of safety requirements management process to the system design:*** Early engagement of product safety team can reduce the lead time due to delays in finalizing the system architecture and safety requirements. Best practice is to engage the safety engineer at the architecture design stage. Safety team involvement and system requirements review before the architecture design is the proposal to eliminate the need for re-baselining the system requirements. Safety management team should be aware of the aircraft level requirements and the system interfaces to aid the fault tolerant architecture design. Safety related requirements validation and verification can be made easy with the usage of system development models. This could be achieved using a common system model shared between system and safety team throughout the development phase.

***Formal review process to finalize the safety requirements:*** Adding a formal internal review during each step of the safety management process is advised to improve the process efficiency. The Peer Review Checklist provides the reviewer with a consistent means of evaluating compliance to the process. A peer review checklist with following considerations should be used to perform an effective peer review. This peer review checklist is a compliment to the safety assessment checklist utilized for performing PSSA review.

- Checkpoints to ensure compliances to standard process
- Checkpoints to evaluate appropriate level of justification for
    a) Rationale for safety requirement and traceability to safety assessment (PSSA, SSA, FMEA, FTA or equivalent)
    b) Rationale for derived requirement defining the need for existence.
- Checkpoints to ensure independency of the derived requirement

The safety requirements emerging from the safety management process should be reviewed and documented as part of the process. Peer review process should ensure flowing the requirements to item level and traceability at lower level for validation and verification, coverage, correctness, and completeness. Traceability of aircraft level safety requirements to the system requirements and then to the item level requirements down to derived safety requirement is necessary. Traceability to the safety analysis as applicable is necessary for a robust rationale in any requirement. This could be established using the linkages to system safety assessment. Furthermore, it is a best practice that the safety engineer is involved in all the requirements review.

***Documenting the non-conformances to safety requirements:*** The requirement management process should make sure proper documentation non compliances to safety requirements. The same should be communicated back to the Aircraft level safety management process.

Safety management plan should establish a communication plan to communicate between the system and aircraft safety team. This eases the complexities arising while managing the safety requirements for complex systems with multiple levels of hierarchy-aircraft/system/item.

# Conclusion

In this paper we have discussed the limitations and issues present in handling the safety requirements during the product development and how it can be mitigated using system models and robust peer review. This paper brings out the need for implementing a matured safety requirement management process as part of the system development. Safety requirements management should always go in hand with the system development process. The discussion included in this paper is based on the ARP4754B process and common industry practices implementing this process.

# Future steps

Industry is moving towards next level enhancement in safety analysis using Model Based System Engineering (MBSE) approach. Model Based Safety Analysis (MBSA) proposes sharing the common system models between system and safety engineering. This approach provides the opportunity to simulate the system behavior during the fault conditions by augmenting the model with a fault model. MBSA based safety analysis provides opportunity to automate some of the analysis steps. But there are associated limitations and challenges. These challenges are not discussed in this paper and are the opportunities to explore further.

# References

ARP-4754B, Guidelines for Development of Civil Aircraft and Systems

ARP, 4761A: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, Society of Automotive Engineers, Inc.

Allenby, K. and Kelly, T. (Washington, 2001), Deriving Safety Requirements using Scenarios. in the 5th IEEE International Symposium on Requirements Engineering (RE'01)

Anjali Joshi, Mats P.E. Heimdahl, Steven P. Miller, Mike W. Whalen, Model-Based Safety Analysis, Department of Computer Science and Engineering, University of Minnesota

Firooz A. Allahdadi, Isabelle Rongier and Paul D. Wilde, Safety Design for Space Operations

Figure 2, ARP4754B, Guidelines for Development of Civil Aircraft and Systems

# Biography

**Manju Maheve**

Manju is a Principal System Safety & Reliability Engineer with over 15 years of experience in aerospace industry. Her expertise spans System Safety Assessment, Reliability Modeling and compliance with standards such as ARP4754B, ARP4761A, DO-178C and DO-254. She has contributed to the development of safety-critical avionics systems. Her current interests focus on model-based safety analysis, AI-driven reliability engineering and advancing digital methods for safety assurance and certification in complex aerospace systems. She holds a Masters ‘degree in Embedded Systems and a bachelor’s degree in Electronics and Communication engineering.