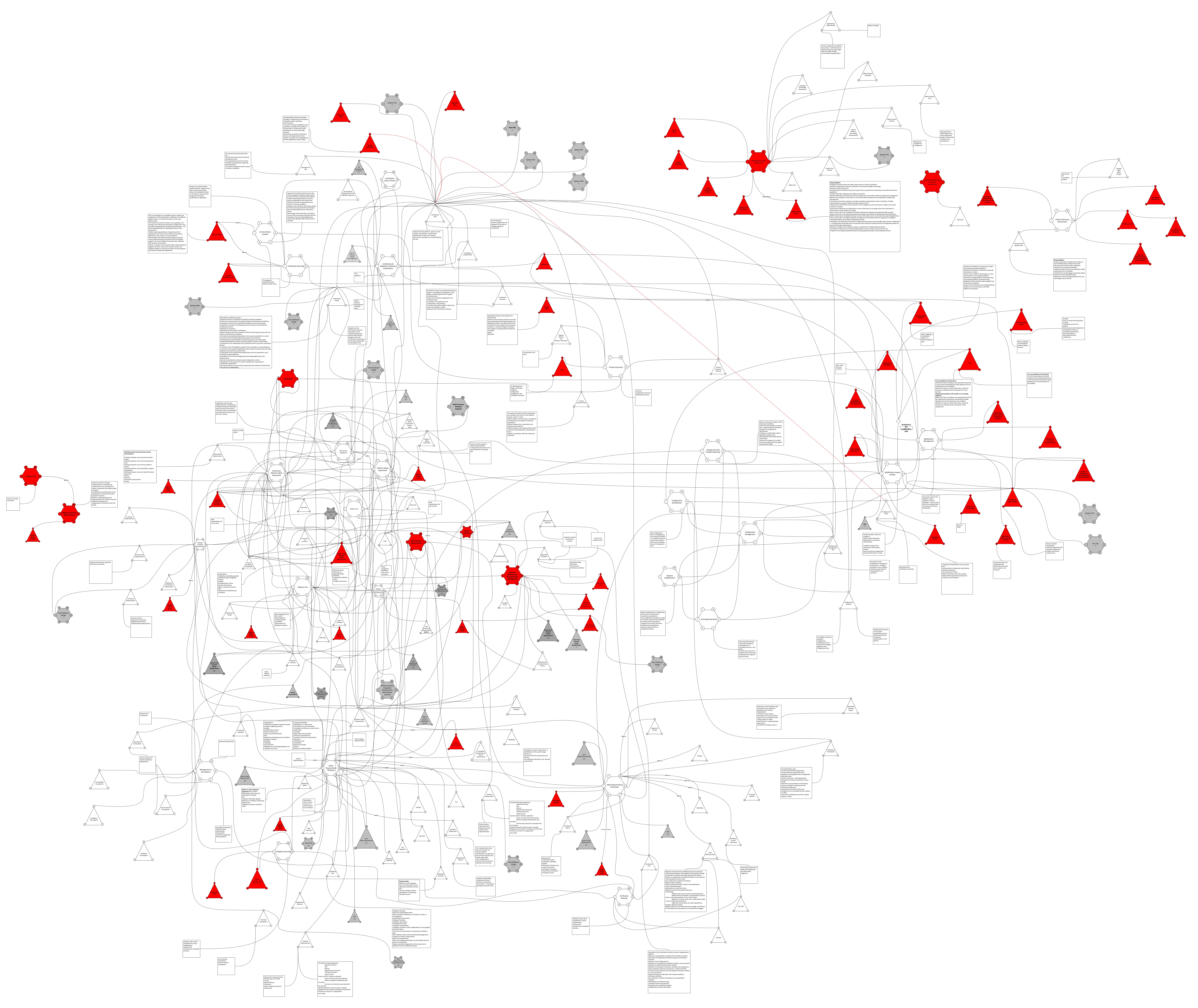

Special Platform-Level Safety Assessment(s)
Safety Lessons Learned
Required by 'programme management'
System FHA
System FHA Manual
Responsibilities:
Establish and communicate the safety requirements at all tiers of definition
Identify aircraft/system functions required for continued safe (flight and landing)
Develop a Platform-level FHA
Develop System FHA Manual and ensure System FHAs are performed consistently in accordance with these guidelines
Conduct thorough, integrated survivability assessments
Identify safety lessons learned from previous programmes and provide visibility to programme management
Monitor the completion of the PSSAs, to ensure that safety requirements are established, validated and documented
Coordinate to promote consistency of analysis methods and approaches used in verification of safety requirements (e.g. mission lengths and other exposure times)
Work with the Chief Pilot to ensure that there are no design issues that could lead to a flight crew caused incident or accident
Work with the Maintenance Organisation to ensure that there are no design issues that could lead to a maintenance-caused incident or accident
Play an active role in the Candidate Certification Maintenance Requirement Meeting (CMR) meetings
Play an active role in the maintenance planning meeting to accomplish the Maintenance Steering Group 3 (MSG-3) planning for scheduled meetings to ensure that safety issues are resolved and properly dispositioned
Play an active role in the Flight Operations development of the Master Minimum Equipment List (MMEL)
Coordinate Platform level Safety Certification activities
Participate in the planning and execution of flight testing (to ensure that the design safety issues are addressed – including determining what tests are required ad the method of testing that will most adequately satisfy the aircraft-level safety requirements
Maintain oversight on test readiness reviews, including the First Flight Readiness Review
Participate in Preliminary and Critical Design Reviews to ensure safety requirements are met.
Prepare for the Engineering Safety Review (reviewing appropriate plans and providing comments

Each zone of the platform after components are situated in the as-built configuration.
Analysis considers the installation aspects of individual systems and components, and the mutual influence between several systems/components in close proximity.
Carried out by 'Zone Chiefs' using participants from the effected design groups and functional support groups

Independence
Probabilistic availability (typically handled through a budgeting process)
Integrity
No single failure criteria
Monitor performance
Safety or protective features
DALs
Operational and Maintenance Limitations

Rationale and/or references to applicable design standards

Carried out at the beginning of the platform development lifecycle

With independence for DAL A and B

Physical attributes of the system in relation to the platform environment

Safety Engineering

Common naming convention
FHA Failure Conditions are adequately addressed

Analyse the effects of design implementation, manufacturing, maintenance errors and failures of system components that defeat design principles.
Considerations should be given to the independence of functions and their respective monitors.
Based on analysing designs and implementation for elements that may defeat the redundancy or independence of functions within the design.

With independence for DAL A and B

Consistent approach to FHAs across the system-level and at platform level

Independence
Probabilistic availability (typically handled through a budgeting process)
Integrity
No single failure criteria
Monitor performance
Safety or protective features
DALs
Operational and Maintenance Limitations

Budgeted probabilistic safety requirements for each system/ function

Uniquely Identifiable
Identifiable as a requirement
Valid (extant and not redundant)
Traceable to arhitecture, source and to lower levels
Verifiable
Measurable and achievable
Measures of performance
Consistent with other requirements
Necessary
Factually correct
Amendable
Rationale provided
Atomic
Reflects the safety analyses

Independence
Probabilistic availability (typically handled through a budgeting process)
Integrity
No single failure criteria
Monitor performance
Safety or protective features
DALs
Operational and Maintenance Limitations
All failure conditions
Reliability
Availability
Fault Tolerance
Mitigation of unsafe design/design errors

Safety Engineering

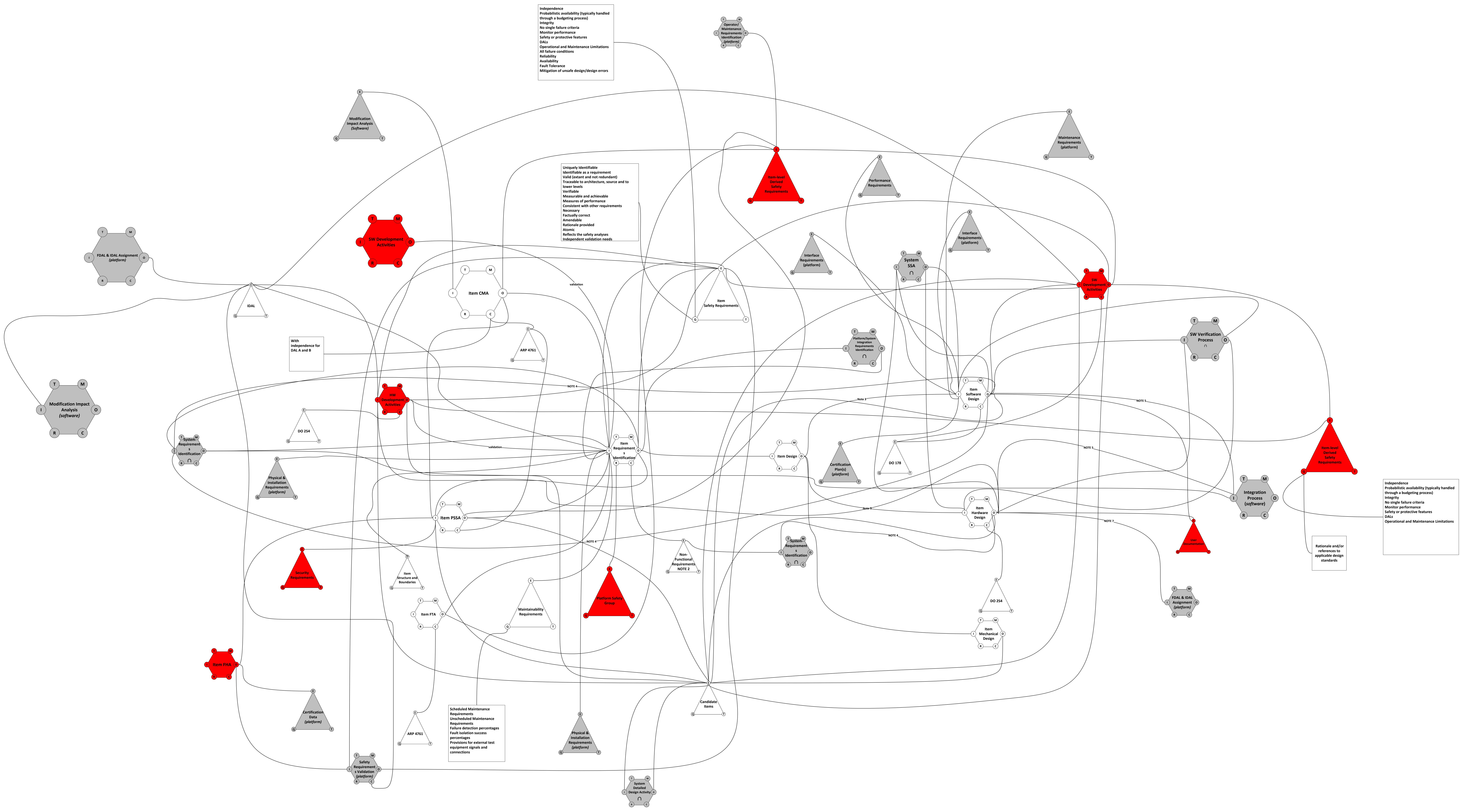
FHA Failure Conditions are adequately addressed

A summary of all failures identified by the FMEA(s) which are grouped together on the basis of their failure effects

Carried out at multiple stages of the system development – including system, component, and hardware/software design definitions

Verify that all significant events identified in the FMES are considered for inclusion as primary events in the FHA

With independence for DAL A and B



Independence
Probabilistic availability (typically handled through a budgeting process)
Integrity
No single failure criteria
Monitor performance
Safety or protective features
DALs
Operational and Maintenance Limitations
All failure conditions
Reliability
Availability
Fault Tolerance
Mitigation of unsafe design/design errors

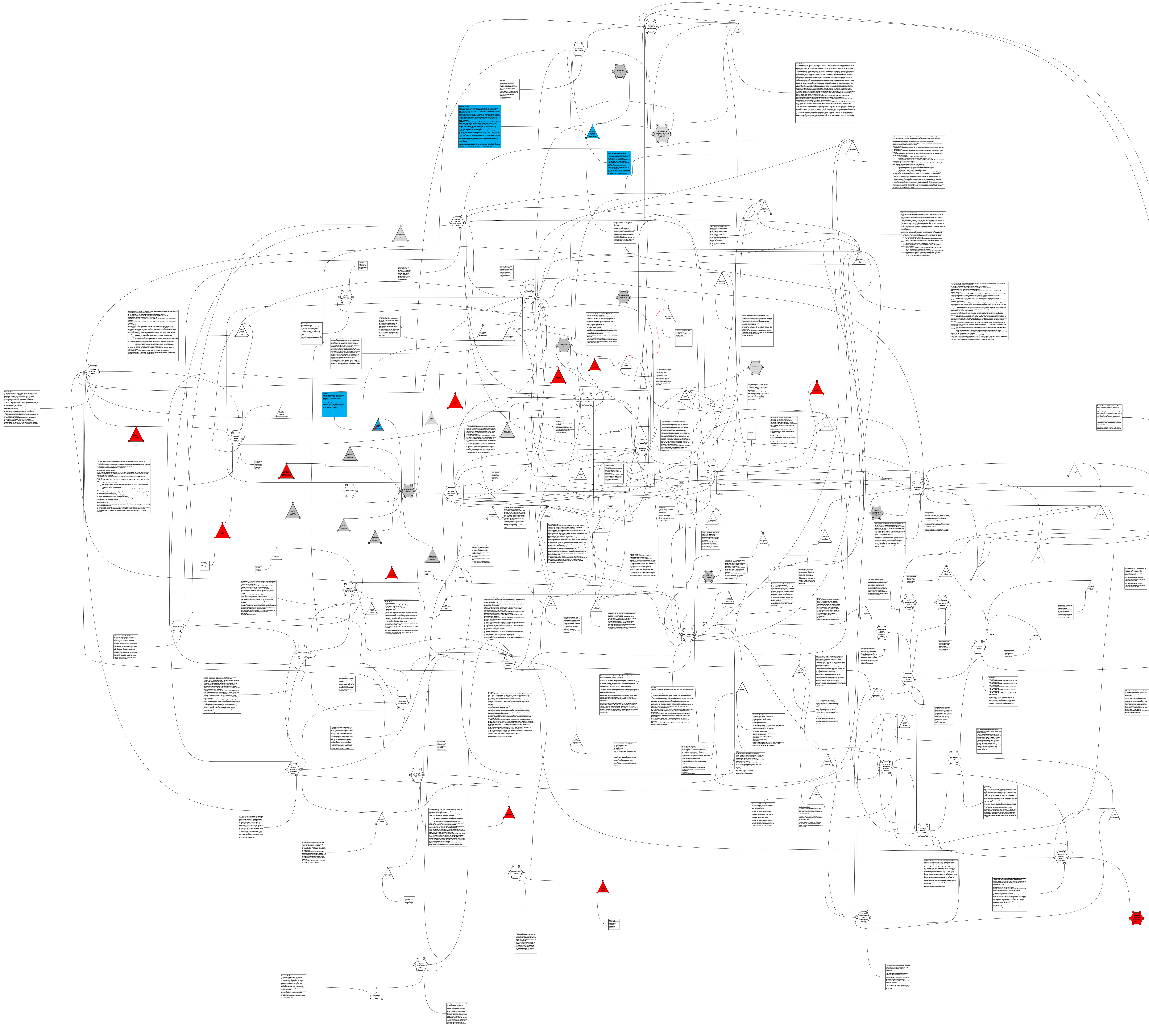
Uniquely Identifiable
Identifiable as a requirement
Valid (extant and not redundant)
Traceable to architecture, source and to lower levels
Verifiable
Measurable and achievable
Measures of performance
Consistent with other requirements
Necessary
Factually correct
Amendable
Rationale provided
Atomic
Reflects the safety analyses
Independent validation needs

With independence for DAL A and B

Scheduled Maintenance Requirements
Unscheduled Maintenance Requirements
Failure detection percentages
Fault isolation success percentages
Provisions for external test equipment signals and connections

Rationale and/or references to applicable design standards

Independence
Probabilistic availability (typically handled through a budgeting process)
Integrity
No single failure criteria
Monitor performance
Safety or protective features
DALs
Operational and Maintenance Limitations



Should include:
1. System Overview: an overview of the system , including a description of its functions and their allocation to hardware and software, the architecture, the processor(s) used, the hardware/software interfaces, and safety features. This section also describes any differences from the system overview in the Plan for Software Aspects of Certification.
2. Software Overview: a description of the SW functions with emphasis on the safety and partitioning concepts used, and explains the differences from the SW overview proposed in the Plan for SW Aspects of Certification.
3. Certification Considerations: restates the certification considerations described in the Plan for Software Aspects of Certification, and describes any differences.
4. Software Lifecycle: summarises the actual software life cycle(s) and explains the differences from the SW lifecycle and SW lifecycle processes proposed in the Plan for Software Aspects of Certification.
5. Software Life Cycle Data: describe the differences from the proposals made in the Plan for Software Aspects of Certification for the SW lifecycle data produced, the relationship of the data to each other and to other data defining the system, and the means by which the data was made available to the certification authority. This section explicitly references, by configuration identifiers and version, the applicable Software Configuration Index and Software Lifecycle Environment Configuration Index. Detailed information regarding configuration identifiers and specific versions of software lifecycle data is provided in the Software Configuration Index.
6. Additional Considerations: summarises any specific considerations that may warrant the attention of the certification authority. It explains any differences from the proposals contained in the Plan for Software Aspects of Certification regarding such considerations. Reference should be made to data items applicable to these matters, such as issue papers or special conditions.
7. Supplier Oversight: describes how supplier processes and outputs comply with plans and standards.
8. Software Identification: identifies the software configuration by part number and version.
9. Software Characteristics: states the EOC size, timing margins, including WCET, memory margins, resource limitations, and the means used for measuring each characteristic/
10. Change History: as applicable, this section includes a summary of SW changes made due to failures affecting safety, and identifies any changes from and improvements to the SW lifecycle processes since the previous certification.
11. Software Status: a summary of Problem Reports unresolved at the time of certification. The Problem Report summary includes a description of each problem and any associated errors, functional limitations, operational restrictions, potential adverse effect(s) on safety together with a justification for allowing the Problem Report to remain open, and details of any mitigating action that has been or needs to be carried out.
12. Compliance Statement: a statement of compliance with DO-178C and a summary of the methods used to demonstrate compliance with criteria specified in the SW plans. This section also addresses additional rulings made by the certification authority and any deviations from the SW plans, standards, and DO-178C not covered elsewhere in the Software Accomplishment Summary.

NOTE 1
The Non-Functional Constraints are: Technology; Schedules; Production Capability; Contracts; Economics; and Experience and Precedence.

NOTE 2
The Non-Functional Requirements are:
Constraints and non-functional requirements; Integration Requirements; and Interface Requirements.

NOTE 3
Requirements allocated to hardware,
Requirements allocated to software,
Development Assurance Levels that are allocated to items,
Description of associated failure conditions,
Allocated failure rates and exposure interval(s) for hardware failures,
System(s) description,
Design constraints,
System verification activities,
Evidence of the acceptability by the system process of any data provided by the hardware and/or software processes to the system process on which any activity/assessment has been conducted by the system process (e.g. the system process evaluation of derived requirements provided by the software process to determine if there is any impact on the SSA).

NOTE 4
Derived requirements for evaluation/validation,
Description of the implemented hardware or software architecture - sufficient to show the achieved independence and fault containment capabilities (e.g. hardware segregation, software partitioning),
Evidence of any system/item verification activities performed at the software or hardware development level,
Hardware failure rates, fault detection coverage, common cause analyses and latency intervals (for incorporation in the SSA),
Problem or change documentation that may impact system or item requirements or hardware/software derived requirements, to be evaluated against the system or item requirements or the safety assessment,
Any limitations of use, configuration identification/status constraints, performance/timing/accuracy characteristics,
Data to facilitate integration of the hardware / software into the system (e.g. installation drawings, schematics, parts lists),
Details of proposed hardware/software verification activities to be performed during system level verification.

NOTE 5
Derived requirements needed for hardware/software integration, such as definition of protocols, timing constraints, and addressing schemes for the interface between hardware and software.

NOTE 6
List of previously agreed upon external event probabilities,
System description including functions and interfaces,
List of Failure Conditions,
Qualitative and quantitative analyses for Failure Conditions (e.g. FTA, FMES, Markov Analysis, Dependence Diagrams),
The results obtained from Common Cause Analyses,
Safety related tasks and intervals (FTA, FMES, Markov Analysis, Dependence Diagrams),
Development Assurance Levels for aircraft functions and systems (FDAL) (PASA, PSSA),
Development Assurance Levels for electronic hardware and software items (IDAL),
The results of the non-analytic verification processes (i.e. test, demonstration and inspection),
A summary of the aircraft safety activities from the beginning of the concept development to the completion of the detailed design development.

NOTE 7
Functional Failure Set, Members, Independence.

NOTE 7b
The output here is specifically considering the assignment of development assurance levels.

NOTE 8
Only if validation of requirements is not possible before design implementation is available.

NOTE 9
Refer to ARP 4754A Table 6 for the recommended methods in accordance with the DAL

NOTE 10
Refer to ARP 4754A Table 6 for the recommended methods in accordance with the DAL

NOTE 11
Original Safety Assessment pertinent to the level of architecture (i.e. system and/or item)

NOTE 12
How service history will be used (e.g. the amount of service experience available and a description of how the service data will be analysed)

NOTE 13
Prior (existing) platform certification

NOTE 14
Those functions required for continued safe flight and landing

NOTE 15
CMA verifies that ANDed gates are truly independent

NOTE 16
CMA verifies that The represents the usual route. Alternative routes are possible – see Fig 3-1 of DO 178C

NOTE 17
When multiple-version dissimilar software is used in a system, the software planning process should choose the methods and tools to achieve the dissimilarity necessary to satisfy the system safety objectives

NOTE 18
Trace data to:
1. Enable verification of the complete implementation of the system requirements allocated to software.
2. Give visibility to those derived high-level requirements that are not directly traceable to system requirements.

NOTE 19
Trace data to:
1. Enable verification that no Source Code implements an undocumented function.
2. Enable verification of the complete implementation of the low-level requirements.

NOTE 20
The need for the requirements to be verifiable once they have been implemented in the software may itself impose additional requirements or constraints on the software development processes.

NOTE 21
Robustness test cases are requirements-based. The robustness testing criteria cannot be fully satisfied if the software requirements do not specify the correct software response to abnormal conditions and inputs. The test cases may reveal inadequacies in the software requirements, in which case the software requirements should be modified. Conversely, if a complete set of requirements exists that covers all abnormal conditions and inputs, the robustness test cases will follow from those software requirements.

NOTE 22
Typical errors revealed by this testing method include:
Incorrect interrupt handling;
Failure to satisfy execution time requirements;
Incorrect software response to hardware transients or hardware failures, e.g. start-up sequencing, transient input loads, and input power transients;
Data bus and other resource contention problems, e.g. memory mapping;
Inability of built-in test to detect failures;
Incorrect behaviour of control loops;
Incorrect control of memory management hardware or other devices under software control;
Stack overflow;
Incorrect operation of mechanism(s) used to confirm the correctness and compatibility of field-loadable software;
Violations of software partitioning.

NOTE 23
Typical errors revealed by this testing method include:
Incorrect initialisation of variables and constants;
Parameter passing errors;
Data corruption, especially global data;
Inadequate end-to-end numerical resolution;
Incorrect sequencing of events and operations.

NOTE 24
Typical errors revealed by this testing method include:
Failure of an algorithm to satisfy a software requirement;
Incorrect loop operations;
Incorrect logic decisions;
Failure to process correctly legitimate combinations of input conditions;
Incorrect responses to missing or corrupted input data;
Incorrect handling of exceptions, such as arithmetic faults or violations of array limits;
Incorrect computation sequence;
Inadequate algorithm precision, accuracy or performance.