



# Integrating System Safety and Software Assurance

Systems Certification and Integrity  
Directorate of Aviation Engineering  
Directorate General Technical Airworthiness



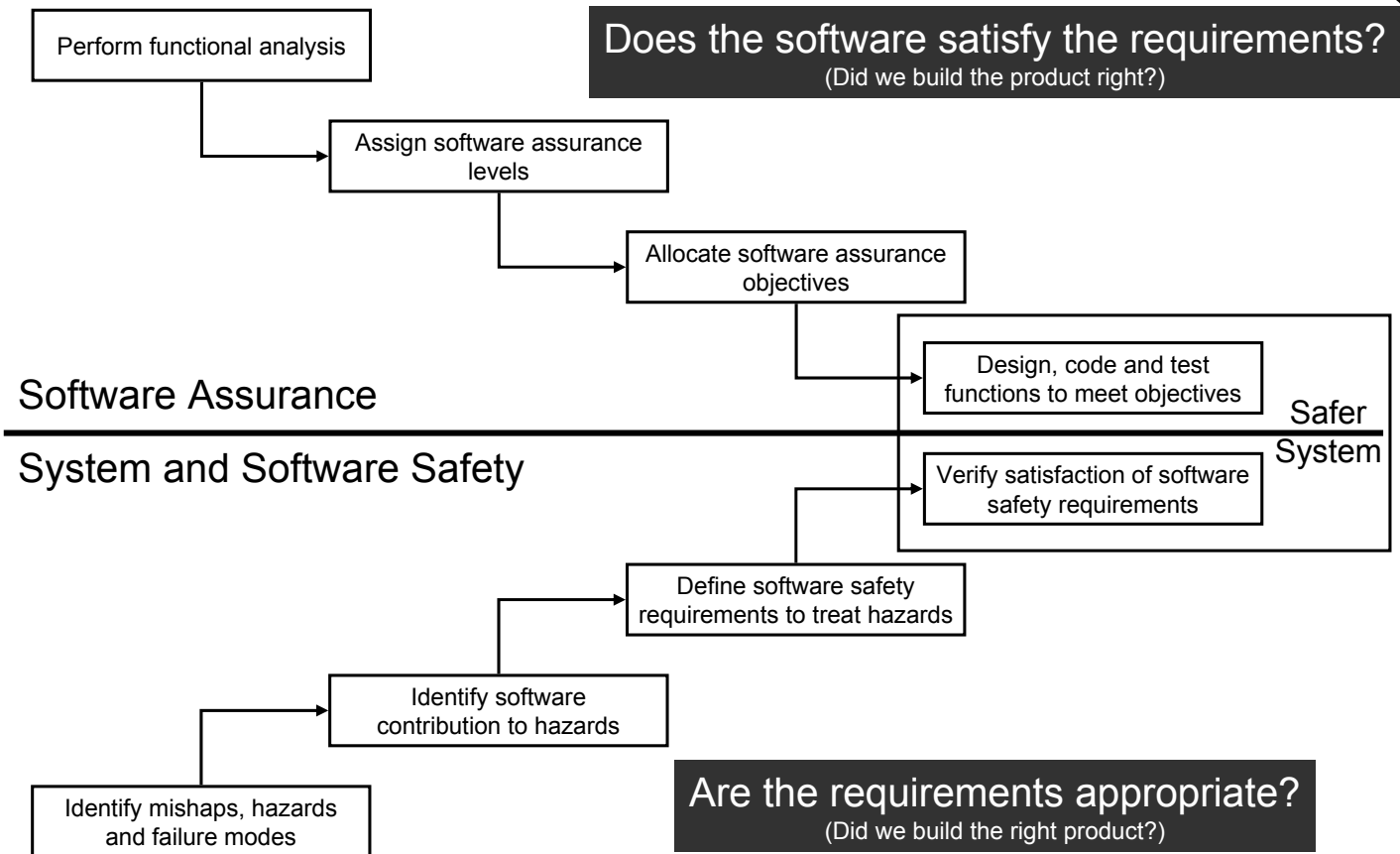
## Overview

- Integration of software assurance into the system safety program.
- Discuss different ADF paradigms for software assurance.
- Explain application of software assurance to missionised hazards and capability integrity.





# Complementary Approaches



# The Goal

- What are we trying to achieve?
  - Establish a standard of proof that software exhibits the required behaviours.
- Which means...
  - What activities do I need to complete in order to prove that my software satisfies its requirements?





# The Basic Concept

The more important it is that the software should work, the more effort should be applied to proving that it does work.



# Software Assurance Paradigms in the ADF

- ARP 4754/4761 & RTCA DO-178B
  - Civil Certified Aircraft
- MIL-STD-882C & RTCA DO-178B
  - Military Aircraft (recent programs)
- MIL-STD-882C & MIL-STD-498/DOD-STD-2167A
  - Military Aircraft (legacy programs)





# ARP 4754/4761 & RTCA DO-178B

Civil Certified Aircraft



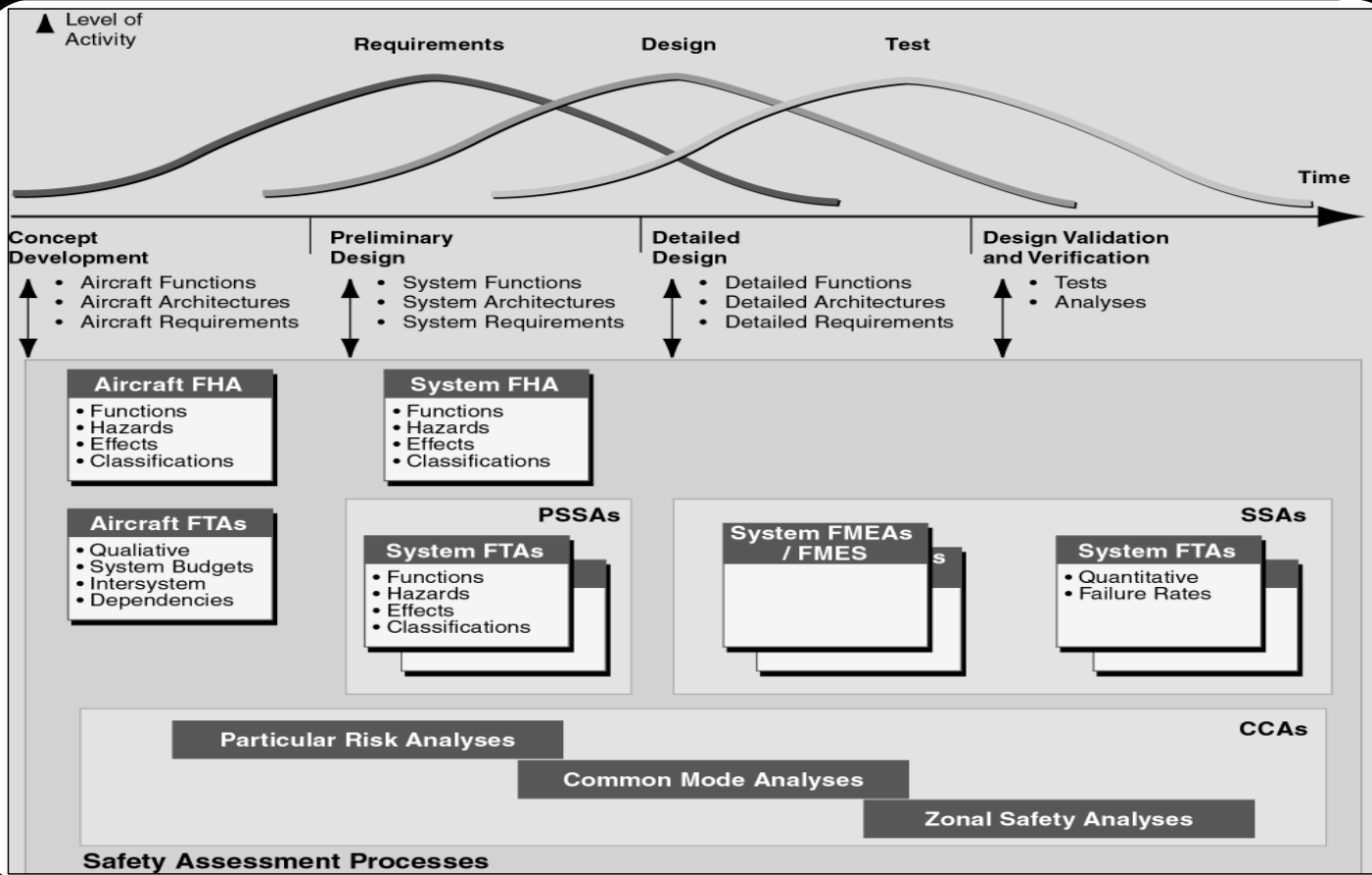
## Applicable Standards

- **FAR 2x.1309**
  - Equipment, Systems and Installations
- **AC 2x.1309**
  - Guidance on complying with FAR 2x.1309
- **SAE ARP 4754 *Certification Considerations for Highly-Integrated or Complex Aircraft Systems***
  - A standard that can be used to demonstrate compliance with FAR 2x.1309
- **SAE ARP 4761 *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment***
  - Describes the techniques that underpin ARP 4754
- **RTCA DO-178B *Software Considerations in Airborne Systems and Equipment Certification***
  - Describes the techniques for treating the software aspects of compliance with ARP 4754





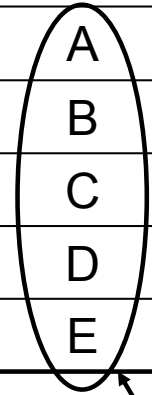
# ARP 4754 System Safety Approach



# ARP 4754 to DO-178B Mapping

## Section 5.4 of ARP 4754 – Design Assurance Determination

Failure Condition Classification	System Development Assurance Level
Catastrophic	A
Hazardous/Severe Major	B
Major	C
Minor	D
No Safety Effect	E



RTCA DO-178B Software Levels

Note:

Section 2.2.2 of DO-178B maps software levels to failure severities, but this is included only for historical reasons (DO-178B was released before ARP 4754). Section 2.2.2 of DO-178B should never be used.





- Section 5.4.1 allows for a reduction in assurance levels for:
  - **Partitioned Designs**
    - Each partition assigned its own level
  - **Dissimilar, Independent Designs Implementing an Aircraft Level Function**
    - Two Level Bs = Level A (but can only use once in any fault tree)
  - **Active/Monitor Parallel Design**
    - One to Level A, the other to Level C
  - **Backup Parallel Design**
    - Primary: Level A, Backup: Level C
  - Note: examples above are for Catastrophic aircraft level failure conditions.



## Example DO-178B Table

Table A-7

	Objective	Ref.	Applicability by SW Level				Output		Control Category by SW level			
			A	B	C	D	Description	Ref.	A	B	C	D
1	Test procedures are correct.	6.3.6b	●	○	○		Software Verification Cases and Procedures	11.13	②	②	②	
2	Test results are correct and discrepancies explained.	6.3.6c	●	○	○		Software Verification Results	11.14	②	②	②	
3	Test coverage of high-level requirements is achieved.	6.4.4.1	●	○	○	○	Software Verification Results	11.14	②	②	②	②
4	Test coverage of low-level requirements is achieved.	6.4.4.1	●	○	○		Software Verification Results	11.14	②	②	②	
5	Test coverage of software structure (modified condition/decision) is achieved.	6.4.4.2	●				Software Verification Results	11.14	②			
6	Test coverage of software structure (decision coverage) is achieved.	6.4.4.2a 6.4.4.2b	●	●			Software Verification Results	11.14	②	②		
7	Test coverage of software structure (statement coverage) is achieved.	6.4.4.2a 6.4.4.2b	●	●	○		Software Verification Results	11.14	②	②	②	
8	Test coverage of software structure (data coupling and control coupling) is achieved.	6.4.4.2c	●	●	○		Software Verification Results	11.14	②	②	②	

LEGEND:

- The objective should be satisfied with independence.
- The objective should be satisfied.
- Blank Satisfaction of objective is at applicant's discretion.
- ① Data satisfies the objectives of Control Category 1 (CC1).
- ② Data satisfies the objectives of Control Category 2 (CC2).



# DO-178B Annex A Tables

Table A.5 – Verification of Outputs of Software Coding and Integration Process

Objective		Applicability by SW Level				Output		Control Category by SW level			
Description	Ref.	A	B	C	D	Description	Ref.	A	B	C	D
Source Code complies with low-level requirements.	6.3.4a	●	●	○		Software Verification Results	11.14	②	②	②	

DO-178B Chapter 6

DO-178B Chapter 11

6.3.4 Reviews and Analyses of the Source Code

The objective is to detect and report errors that may have been introduced during the software coding process. These reviews and analyses confirm that the outputs of the software coding process are accurate, complete and can be verified. Primary concerns include correctness of the code with respect to the software requirements and the software architecture, and conformance to the Software Code Standards. These reviews and analyses are usually confined to the Source Code. The topics should include:

a. Compliance with the low-level requirements: The objective is to ensure that the Source Code is accurate and complete with respect to the software low-level requirements, and that no Source Code implements an undocumented function.

11.14 Software Verification Results

The Software Verification Results are produced by the software verification process activities. Software Verification Results should:

- For each review, analysis and test, indicate each procedure that passed or failed during the activities and the final pass/fail results.
- Identify the configuration item or software version reviewed, analyzed or tested.
- Include the results of tests, reviews and analyses, including coverage analyses and traceability analyses.



## Is it really that easy?

- Hazardous/Severe Major
  - Failure conditions that could result in a large reduction in safety margins.
- Major
  - Failure conditions that could result in a significant reduction in safety margins.
- What is the difference between large and significant?





# How does SCI-DGTA handle this?

- Look to FAA policy: ACs, TSOs, etc
- Best Reference: AC 23.1309-1D Appendix 1

Aircraft Function	Classification of Failure Conditions		
	Total Loss of Function	Loss of Primary Means of Providing Function	Misleading and/or Malfunction Without Warning
Display of attitude information to control roll and pitch	Catastrophic	Major	Catastrophic

- No published list for Part 25 (high value IP)
- Another Example: TSO-C151b TAWS
  - “Software implementing the functions defined in this TSO must be developed to Level C as defined in RTCA DO-178B.”



# MIL-STD-882C & RTCA-DO-178B

## Military Aircraft (recent programs)





# Integrating Software Assurance into Military System Safety Engineering

- **Step 1:** Functional Safety Analysis/Assessment
- **Step 2:** Categorisation of each software function IAW Safety Integrity Level (SIL) definitions
- **Step 3:** Providing the software development team with the required tasks for software development and test for each SIL (i.e. process objectives).
- **Step 4:** Implement the software development and test tasks in accordance with the SIL definitions.



# MIL-STD-882C is Task Based

Task	Title	Task Type
101	System Safety Program	MGT
102	System Safety Program Plan	MGT
103	Integration/Management of Associate Contractors etc	MGT
104	System Safety Program Review/Audits	MGT
105	SSG/SSWG Support	MGT
106	Hazard Tracking and Risk Resolution	MGT
107	System Safety Progress Summary	MGT
201	Preliminary Hazard List	ENG
202	Preliminary Hazard Analysis	ENG
203	Safety Requirements/Criteria Analysis	ENG
204	Subsystem Hazard Analysis	ENG
205	System Hazard Analysis	ENG
206	Operating and Support Hazard Analysis	ENG

Note that there are no software specific tasks!





## Sample Hazard Severity Categories (MIL-STD-882C)

Description	Cat	Definition
Catastrophic	I	Death, system loss or severe environmental damage.
Critical	II	Severe injury, severe occupational illness, major system or environmental damage.
Marginal	III	Minor injury, minor occupational illness or minor system or environmental damage.
Negligible	IV	Less than minor injury, occupational illness or less than minor system or environmental damage.



## Sample Hazard Probability Levels (MIL-STD-882C)

Description	Cat	Specific Individual Item	Fleet or Inventory
Frequent	A	Likely to occur frequently	Continuously experienced
Probable	B	Will occur several times in the life of an item	Will occur frequently
Occasional	C	Likely to occur some time in the life of an item	Will occur several times
Remote	D	Unlikely but possible to occur in the life of an item	Unlikely but can reasonably be expected to occur
Improbable	E	So unlikely it can be assumed occurrence may not be experienced	Unlikely to occur, but possible





# Sample Hazard Risk Index Matrix (MIL-STD-882C)

	CAT	CRIT	MAR	NEG
<b>Frequent</b>	1	3	6	10
<b>Probable</b>	2	5	8	12
<b>Occasional</b>	4	7	11	14
<b>Remote</b>	10	13	16	18
<b>Improbable</b>	15	17	19	20

Colours link to risk acceptance authority (e.g. ADF AA, OAA, OAAR, DAR).

Likelihoods cannot be correctly assigned to software causal factors: how then does software fit into the above matrix?



# Software Control Categories

Control Category	Description
Ia	Direct control.
Ib	Displays information that is directly and constantly relied upon.
IIa	Direct control but time for intervention by independent safety system.
IIb	Displays information that if incorrect, may lead to operator action/inaction that progressively degrades safety margins.
IIIa	Control over hazardous systems, but operator action required to complete the task.
IIIb	Displays information about discrete events that require an operator to respond to prevent a hazard occurring.
IV	No control of safety related functions.

Summary of Definitions from 7001.054 Section 2 Chapter 7.





# Software Hazard Risk Index

		Hazard Category			
		CAT	CRIT	MAR	NEG
Control	I	1	2	3	4
	II	2	3	4	4
	III	3	4	4	4
	IV	5	5	5	5

Aircraft hazards will mostly fall into here...

SHRI	DO-178B Level
1	A
2	B
3	C
4	D
5	E

Satisfaction of assurance level is the measure of hazard acceptability.



# SHRI vs. HRI

- SHRIs and HRIs are not interchangeable or comparable.
- Consider multi-axis aircraft stability augmentation system.
  - Loss of function Catastrophic.
- Probabilistic (hardware) Failure Modes:
  - High Probability ( $> 10^{-2}$ ): HRI = 1
  - Low Probability ( $< 10^{-9}$ ): HRI = 15
  - HRI is a measure of acceptability
- Systematic (software) Failure Modes
  - Poor Software: SHRI = 1 (Catastrophic/Control Category Ia)
  - Good Software: SHRI = 1 (Catastrophic/Control Category Ia)
  - SHRI is not a measure of acceptability.





# Don't Confuse Methods

- Navigation System for Large Aircraft
- ARP 4754 & DO-178B
  - Loss of: Major
  - Malfunction: Hazardous
  - Requires DO-178B Level B
- MIL-STD-882C & DO-178B
  - Hazard Category: Catastrophic (CFIT)
  - Control Category: IIb (Display of Information)
  - SHRI 2: Requires DO-178B Level B
- Incorrect
  - Malfunction: Hazardous → Critical in 882C terminology
  - Control Category: IIb
  - SHRI 3: Requires DO-178B Level C



# MIL-STD-882C & MIL-STD-498/DOD- STD-2167A

Military Aircraft (legacy programs)





# MIL-STD-882C to MIL-STD-498 Mapping

		Hazard Category			
		CAT	CRIT	MAR	NEG
Control	I	1	2	3	4
	II	2	3	4	4
	III	3	4	4	4
	IV	5	5	5	5

SHRI	MIL-STD-498 Level
1	??
2	??
3	??
4	??
5	??

MIL-STD-498 does not define varying levels of development and verification rigour.



# Software Assurance Matrix

Phase \ SHRI	Requirements	Design	Code	Test
1 High Risk			All executable object code is directly attributable to the source code.	<ul style="list-style-type: none"> <li>- During testing, each condition has taken both possible values.</li> <li>- During testing, each condition was shown to independently affect the outcome of the decision.</li> </ul>
2 Serious Risk	<ul style="list-style-type: none"> <li>- Software requirements are compatible with target computer.</li> </ul>	<ul style="list-style-type: none"> <li>- Software design and architecture are compatible with the target computer.</li> </ul>		<ul style="list-style-type: none"> <li>- During testing, each decision has taken both possible outcomes.</li> </ul>
3 Medium Risk	<ul style="list-style-type: none"> <li>- Software requirements conform to standards.</li> </ul>	<ul style="list-style-type: none"> <li>- Software design and architecture are defined.</li> <li>- Software design and architecture are accurate, consistent, conform to standards and comply with software requirements.</li> <li>- Traceability between software requirements and software design is established.</li> </ul>	<ul style="list-style-type: none"> <li>- Source code is accurate, consistent, conforms to standards and complies with the software design and architecture.</li> <li>- Traceability between software design and source code is established.</li> </ul>	<ul style="list-style-type: none"> <li>- Software is tested against each software design element under normal and abnormal conditions.</li> <li>- During testing, each line of source code was executed.</li> </ul>
4 Low Risk	<ul style="list-style-type: none"> <li>- Software requirements are defined.</li> <li>- Software requirements are accurate, consistent and comply with system requirements.</li> <li>- Traceability between system requirements and software requirements is established.</li> </ul>	Nil	<ul style="list-style-type: none"> <li>- Source code was developed.</li> <li>- Executable object code was produced.</li> </ul>	<ul style="list-style-type: none"> <li>- Software was tested against each software requirement under normal and abnormal conditions.</li> <li>- Key functions and functions that rely heavily on hardware behaviours were tested on the target computer.</li> </ul>

## AAP 7001.054 Section 2 Chapter 7



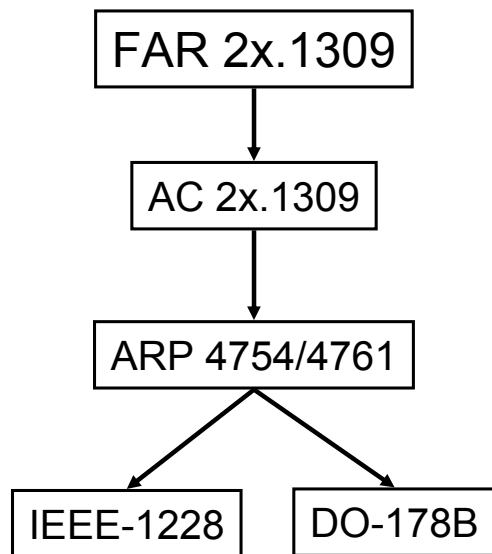
From MIL-STD-882D Rev 1

	Criticality Rating			
	(1) High	(2) Serious	(3) Medium	(4) Low
<b>General</b>				
Peer reviews of all development artifacts are conducted at each phase (requirements, design, code, and test).	M	M	R	R
All design and software components containing safety critical functionality are identified as safety critical and linked to the appropriate software requirements specification (SRS) requirement(s).	M	M	R	R
All SCFs and components are documented and linked to the individual hazards identified in the hazard analysis	M	M	M	R
<b>Requirements Analysis Phase</b>				
Independent analysis/verification of algorithms, limits, ranges, critical values, rate, units, frequency, and volume via independent evaluation.	M	R	R	NR
Traceability of safety critical requirements from hazard analyses to SRS, software design, code, and test.	M	M	M	R
All safety-critical software requirements are broken down to their lowest level and linked to their higher level requirement.	M	M	R	R
All safety-critical software requirements are analyzed for verifiability, testability, and potential confliction with other requirements.	M	M	R	R
All safety-critical requirements are evaluated for timing, resource utilization, and throughput.	M	M	M	R
Use defined safety-related requirements guidelines.	M	M	R	NR
<b>Architectural and Detailed Design Phase</b>				
Evaluate safety-related components for reliability, maintainability, understandability, and performance.	M	M	M	R
Peer reviews of software units identified as safety critical will require the attendance of a reviewer independent from the Software Development Team.	M	M	M	R
For all safety-critical requirements, ensure that there are no common cause failures between components (i.e., FTA).	M	M	R	R
For all safety-critical components, identify any or all dependencies.	M	M	R	R
Verify accuracy and correctness of all algorithms in safety-critical components.	M	M	M	M
Verify all data used in safety-critical components are used as specified and are consistently used between components.	M	M	M	R
Verify all interfaces between safety-critical components.	M	M	M	R
Evaluate feasibility of safety-critical design constraints.	M	M	R	R
Evaluate partitioning of safety-critical software (goal is to minimize the number of safety-critical components).	M	M	M	R
<b>Coding Phase</b>				
Conduct a safety-critical function code review (with safety engineering attendance)	M	M	R	R
Conduct an independent verification of safety-critical algorithms for accuracy, correctness, and boundary values; data consistency between components; and use defined safety-critical guidelines.	M	M	M	R
Use software coding standards that require software units satisfy the safety-critical requirements	M	M	M	M
Evaluate safety-critical software units for logic, data, and interface errors.	M	M	M	M
Analyze and test algorithms and mathematical computations for accuracy and correctness.	M	M	M	M

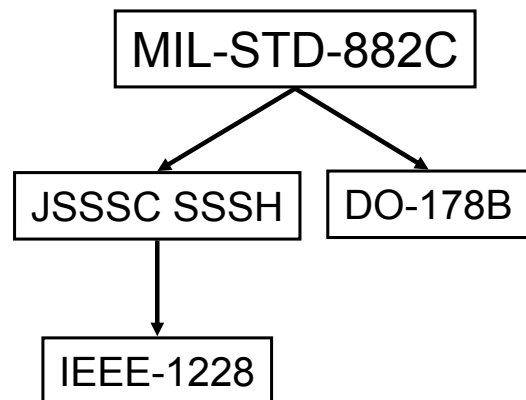


# Putting it All Together

## Commercial Aircraft



## Military Aircraft



These are the ADF preferred airworthiness standards for aviation software.





# Weapons STANAG 4404 and AOP 52



## Background

- DEOP 102 approves the following software standards:
  - AOP-52
    - Guidelines for the Safety and Suitability for Service of Safety Critical Computing Systems in Munitions Applications
  - DEF (AUST) 5679
    - The Procurement of Computer-Based Safety Critical Systems
  - IEEE/EIA STD 12207
    - Software Life Cycle Processes – Implementation Considerations
  - RTCA/DO-178B
    - Software Considerations in Airborne Systems and Equipment Certification
  - STANAG 4404
    - Safety Design Requirements and Guidelines for Munitions Related Safety Critical Computing Systems
  - STANAG 4452
    - Safety Assessment Requirements for Munition Related Computing Systems





# STANAG 4404

- STANAG 4404 combines software safety and software assurance considerations.
- It identifies:
  - Generic software safety requirements.
  - Good software design practices.
  - Verification benchmarks.
- Apply practices to Safety Critical Computing System Functions (SCCSF).
  - Provides guidance on which functions are likely to be SCCSFs.
- Assigns example MIL-STD-882C control categories to weapon functions.



# STANAG 4404 Requirements Applicability

Table B-1  
Design Requirement and Guideline Tailoring  
System Type/Guideline Applicability

Design Guideline Reference	F&E	S	GS	T&L	P&F	C&C
<b>6. Design and Development Process</b>						
6.1 Configuration control	R	R	R	R	R	R
6.2 Quality Assurance Program	R	R	R	R	R	R
6.3 Two person rule	R	R	R	R	R	R
6.4 Reviews and audits	R	R	R	R	R	R
6.5 Program patch prohibition	R	R	R	R	R	R
6.6 Design Verification/Validation	R	R	R	R	R	R
<b>7. System Design Requirements</b>						
7.1 Designed safe states.	R	R	R	R	R	R
7.2 Standalone computer	O	O	O	O	O	O
7.3 Ease of maintenance	R	R	R	R	R	R
7.4 Safe state return	N	N	O	R	R	R
7.5 Restoration of interlocks	N	N	N	R	R	R
7.6 Input/Output registers	O	O	R	R	R	R
7.7 External hardware failures	O	O	O	R	R	R
7.8 Safety kernel failure	R	R	R	R	R	R
7.9 Circumvent unsafe conditions	N	N	N	R	R	R
7.10 Fallback and recovery	N	N	N	R	R	R
7.11 Simulators	R	R	R	R	R	R
7.12 System error log	N	N	N	R	R	R
7.13 Positive feedback mechanisms	R	R	R	R	R	R

**F&E:** Fuzes, Electronic Arming and Safety Devices

**S:** Seekers

**GS:** Guidance Systems

**T&L:** Targeting and Launcher Control Systems

**P&F:** Pointing and Firing Cutout Systems

**C&C:** Command and Control Systems





# Using STANAG 4404

- If STANAG 4404 is to be used, there are a number of shortfalls that should be resolved:
  - STANAG 4404 was never ratified, so it doesn't really exist.
    - SCI-DGTA is aware of at least two different versions.
    - You may need to apply your own configuration control, or include a copy in the contract.
  - Some requirements are ambiguous.
    - Example: All software testing shall be controlled by a formal test coverage analysis and document. Computer based tools shall be used to ensure that the coverage is as complete as possible.
    - When is coverage complete? Statement? Decision? Condition/Decision? Exhaustive?
    - What is 'as complete as possible'?
    - You may need to provide clarification to some requirements.



# AOP-52

- AOP-52 replaced STANAG 4404 and STANAG 4452.
- Strikingly similar to the JSSSC SSSH.
- AOP-52 is a guidance document.
  - It does not contain clear requirements.
  - It would be very difficult, if not impossible, to apply it under a contract.
  - But, it does contain good information and practices.
    - You might be able to convert it to an enforceable standard.





# Using STANAG 4404 – An Example

- Consider a weapon guidance system.
- Is it a Safety Critical Computing System Function?
  - Probably: see para 5.3.a
    - “Any function which determines, controls, or directly influences the flight path of a munition system.”
  - Annex B: A guidance system with a self destruct capability is Control Category II:
    - “Software exercises control over potentially hazardous hardware systems, subsystems, or components allowing time for intervention by independent safety systems to mitigate the hazard. However, these systems by themselves are not considered adequate.”
- Which requirements are applicable?
  - See GS column of Table B-1.



# Missionised Hazards and Capability Integrity





# The Basic Concept

The more important it is that the software should work, the more effort should be applied to proving that it does work.

.... except now the motivation is completion of the mission.



# Definitions

- Missionised Hazards
  - ... aircraft system hazards considered not only within benign operating environments but also within worst credible missionised scenarios.
- Capability Integrity
  - ... functionality required for achieving mission objectives but with limited airworthiness impact.





# Missionised Hazards

- May be considered as part of the functional and system-level hazard assessments.
  - Operator input at SSWGs.
- Tracked and maintained through the hazard log.
- Requires good engineering judgement
  - credible scenarios
- Examples???
  - Not used very often.



# Capability Integrity Definitions

- **Mission Critical**
  - Failure conditions would immediately raise the threat posed by external events (e.g. enemy action) to unacceptable levels and could result in loss of the aircraft.
- **Mission Serious**
  - Failure conditions would reduce the capability of the aircraft or the ability of the crew to cope with adverse mission conditions to the extent that there would be:
    - a large reduction in mission capabilities,
    - higher workload such that the flight crew could not be relied upon to perform their tasks accurately or completely, or
    - increased risk to aircrew and occupants.
- **Mission Degraded**
  - Anything else.





# Software Level Assignments

AAP 7001.054 Section 2 Chapter 7

Mission Failure Categorisation	Software Level Assignment
Mission Critical	DO-178B Level C
Mission Serious	DO-178B Level D
Mission Degraded	No additional objectives

DO-178B sets objectives for developing and verifying high-integrity software. The need for high-integrity software can be driven by either safety or mission considerations.



# Capability Integrity – Examples

Mission System Function	Mission Failure Categorisation	Considerations
RWR/MWS/CMDS	Mission Critical	For aircraft that operate in a hostile environment (e.g. F/A-18)
RWR/MWS/CMDS	Mission Serious	For aircraft that operate in a more benign environment (e.g. AAR)
ESM	Mission Serious	Large reduction in mission capabilities, but no direct threat to the aircraft.
Data Link	Mission Degraded	Mission more difficult, but still possible.





# Capability Integrity

- Clearly articulated in the contract
  - e.g. AAR to Level C, AP-3C ESM to Level D
- Mission assumptions verified by operators
  - Input at SSWGs
  - Documented
- Requires good engineering judgement
  - credible scenarios



# Summary

- Three paradigms for integrating software assurance into the system safety program.
  - Civil: Severity of failure linked to assurance level.
  - Military: Hazard Category and Control Category.
  - Both: more safety critical → more effort.
- Missionised Hazards
- Capability Integrity





# Questions

