Software Certification under the DASRs

Software Symposium 2019
MAJ Pudadera
02 Aug 19
Scope

- Historical Analysis
  - TAREGs (Sep 15)
  - ADRM (2012)

- Current Environment
  - DASRs
    - eADRM (Mar 19)

- Future Direction
  - DASRs
  - ADRM (Q4 19?)
TAREGs

• Two software-centric regulations
  – 2.2.11 Compliance Findings
  – 3.5.3 Software Integrity Management
2.2.11. Compliance Findings

a. The DAR must ensure that:

1. compliance findings are made only by competent and authorised individuals and agencies, and
2. authorisations are recorded within the Design Acceptance strategy.

b. Compliance findings for an ADF SOR requirement must be made on the basis of:

1. data obtained through inspection, test or analysis of the aircraft, its systems, materials, or parts where such tests are conducted subsequent to the relevant design agency tests;
2. data obtained through involvement in design agency inspection or test activity; and
3. Type Design data provided by the design agency, including data related to inspections, tests or analysis conducted without ADF involvement; and/or
4. evidence of prior acceptance, as required by TAREG 2.2.7.

c. An authorised person or agency must only make a compliance finding if:

1. the evidence provided indicates that the design satisfies the ADF SOR requirement, and
2. the method used to show compliance is suitable for its intended purpose.

d. The DAR must ensure that all data assessed in the making of a compliance finding is the applicable version of such data.

e. The DAR must ensure that any issue identified as a result of compliance finding activities is only closed after:

1. the design is subsequently found to meet the relevant ADF SOR requirement,
2. the TAR or a relevant ASR has determined that the design provides an equivalent level of safety to that specified in the ADF SOR,
3. the TAR or a relevant ASR has accepted the reduction in the level of safety where the design has not met all airworthiness standards specified in the ADF SOR, or
4. closure of an Airworthiness Issue Paper raised to communicate the issue to the TAR and OAA.

f. The DAR must ensure that software compliance findings are conducted in accordance with a TAR approved software compliance finding plan where aviation software contributes to a hazard equal to or greater than 'CRITICAL' (MIL-STD-882) or 'HAZARDOUS' (2x.1309) severity.
3.5.3. Software Integrity Management

a. **Software Safety**: Where aviation software contributes to a hazard, software safety must be achieved through:
   1. the generation of requirements specifying safe behaviour; and
   2. the establishment of a development environment that seeks to avoid the introduction of unexpected behaviours.

b. **Software Assurance**: Aviation software must be assured to perform to its specified functions, to provide confidence that performance will be as intended and safe.

c. **Certification Liaison**: The plan for certifying aviation software that contributes to a hazard, equal to or greater than 'CRITICAL' (MIL–STD–882) or 'HAZARDOUS' (2x.1309) severity must be approved by the TAR.

d. **Software Load Control**: Controls must be established to ensure the aviation software configuration is:
   1. approved for installation;
   2. installed in the required configuration; and
   3. verified to be installed correctly.

e. **Software Problem Reporting**: Anomalous behaviour of aviation software must be:
   1. identified and tracked; and
   2. subjected to risk management where unsafe behaviour results in hazards.
Current Environment - DASR

- No Software-specific regulations
  - Green text emphasis on Software
- Software under multiple locations
- Software mostly to be treated as per any other technology
  - Shifts responsibility to Design Org to manage SW changes
## Current Environment - DASR

### TAREG 3.5.3 – Software Integrity Management System

#### a. Software Safety:
Where aviation software contributes to a hazard, software safety must be achieved through:

1. the generation of requirements specifying safe behaviour; and
2. the establishment of a development environment that seeks to avoid the introduction of unexpected behaviours.

<table>
<thead>
<tr>
<th>TAREG 3.5.3 – Software Integrity Management System</th>
<th>DASR coverage of TAREG</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Software Safety: Where aviation software contributes to a hazard, software safety must be achieved through:</td>
<td>DASR 21 Regulations:</td>
</tr>
<tr>
<td>1. the generation of requirements specifying safe behaviour; and</td>
<td>Subpart J – Military Design Organisation Approval</td>
</tr>
<tr>
<td>2. the establishment of a development environment that seeks to avoid the introduction of unexpected behaviours.</td>
<td>21.A.239 - Design assurance system and Safety Management System</td>
</tr>
</tbody>
</table>

#### b. Software Assurance:
Aviation software must be assured to perform to its specified functions, to provide confidence that performance will be as intended and safe.

<table>
<thead>
<tr>
<th>DASR coverage of TAREG</th>
</tr>
</thead>
</table>

#### c. Certification Liaison:
The plan for certifying aviation software that contributes to a hazard, equal to or greater than 'CRITICAL' (MIL–STD–882) or 'HAZARDOUS' (2x.1309) severity must be approved by the TAR.

<table>
<thead>
<tr>
<th>DASR 21 Regulations:</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.A.20 - Compliance with the type-certification basis and environmental protection requirements</td>
</tr>
<tr>
<td>21.A.91 - Classification of changes in type design</td>
</tr>
<tr>
<td>21.A.97 – Major Changes</td>
</tr>
</tbody>
</table>
Current Environment - DASR

21.A.239 - Design assurance system and Safety Management System

(a) The design organisation shall demonstrate that it has established and is able to maintain a design assurance system for the control and supervision of the design, and of design changes, of products, parts and appliances covered by the application. This design assurance system shall be such as to enable the organisation: ▶ GM1 ▶ GM2

1. To ensure that the design of the products, parts and appliances or the design change or repair solution thereof, comply with the applicable type-certification basis and environmental protection requirements (where applicable); and

2. To ensure that its responsibilities are properly discharged in accordance with:

   i. The appropriate provisions of this DASR; and

   ii. The terms of approval issued under DASR 21.A.251.

3. To independently monitor the compliance with, and adequacy of, the documented procedures of the system. This monitoring shall include a feed-back system to a person or a group of persons having the responsibility to ensure corrective actions. ▶ AMC

(b) The design assurance system shall include an independent checking function of the showings of compliance on the basis of which the organisation submits compliance statements and associated documentation to the Authority. ▶ AMC

(c) The design organisation shall specify the manner in which the design assurance system accounts for the acceptability of the parts or appliances designed or the tasks performed by partners or subcontractor according to methods which are the subject of written procedures. ▶ GM

(d) The organisation shall establish and maintain a Safety Management System, in accordance with DASR SMS.
Current Environment - DASR

TAREG 3.5.3 – Software Integrity Management System

\textbf{d. Software Load Control:} Controls must be established to ensure the aviation software configuration is:

1. approved for installation;
2. installed in the required configuration; and
3. verified to be installed correctly.

DASR coverage of TAREG

\textbf{DASR 21 Regulations:}
- Subpart F – 21.A.129 – Obligations of the manufacturer
- Subpart G – 21.A.165 – Obligations of the holder
- Subpart K – 21.A.30x – Parts and Appliances
- Subpart Q – 21.A.80x – Identification of products, parts, and appliances

\textbf{DASR M Regulations:}
- M.A.201 – Responsibilities

\textbf{DASR 145 Regulations:}
- 145.A.42 – Acceptance of components
- 145.A.48 – Performance of Maintenance
Current Environment - DASR

<table>
<thead>
<tr>
<th>TAREG 3.5.3 – Software Integrity Management System</th>
<th>DASR coverage of TAREG</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>e. Software Problem Reporting:</strong> Anomalous behaviour of aviation software must be:</td>
<td><strong>DASR 21 Regulations:</strong></td>
</tr>
<tr>
<td>1. identified and tracked; and</td>
<td>21.A.3A – Failures, Malfunctions and defects</td>
</tr>
<tr>
<td>2. subjected to risk management where unsafe behaviour results in hazards.</td>
<td>21.A.129 - Obligations of the manufacturer</td>
</tr>
<tr>
<td></td>
<td>21.A.165 – Obligations of the holder</td>
</tr>
<tr>
<td></td>
<td><strong>DASR M Regulations:</strong></td>
</tr>
<tr>
<td></td>
<td>M.A.202 – Occurrence reporting</td>
</tr>
<tr>
<td></td>
<td><strong>DASR 145 Regulations:</strong></td>
</tr>
<tr>
<td></td>
<td>145.A.60 – Occurrence reporting</td>
</tr>
<tr>
<td></td>
<td>145.A.95 – AMO Findings by the NMAA</td>
</tr>
</tbody>
</table>
Current Environment - DASR

<table>
<thead>
<tr>
<th>TAREG 2.2.11.f – Software Compliance Findings:</th>
<th>DASR coverage of TAREG 21 Regulations:</th>
</tr>
</thead>
<tbody>
<tr>
<td>The DAR must ensure that software compliance findings are conducted in accordance with a TAR approved software compliance finding plan where aviation software contributes to a hazard equal to or greater than 'CRITICAL' (MIL–STD–882) or 'HAZARDOUS' (2x.1309) severity.</td>
<td>DASR 21.A.20.(b) – Certification Programme</td>
</tr>
</tbody>
</table>
ADRM (2012)

- Highly Prescriptive
  - 22 Core Pages
  - 4 Annexes
CHAPTER 7 AVIATION SOFTWARE

Introduction
Scope and Applicability
Software Safety
  Software Safety Program
  Interaction of the Software Safety Program with the System Safety Program
  Software Safety Requirements and the Safety Case
  Capability Maturity vs Software Safety
  Integrating Software Assurance Requirements into the Software Safety Process
Software Assurance
  Software Assurance Standards
  Software Assurance Products
  CMMI vs. Software Assurance
  Missionised Hazards and Capability Integrity
Software Engineering (Life Cycle and Measurement)
  Software Development Capability
  Software Life-cycle and Development Standards
  Reviews and Audits
  Metrics
  Independent Verification and Validation (IV & V)
Interaction Between System Safety, Software Safety, Software Assurance and Software Development
Software Acquisition
Software Transition to In-Service Support
ADRM (2012)

AAP 7001.054(AM1)

SECTION 2 (cont)

CHAPTER 7 (cont)

Transition of Standards
Deliverable Requirements
Intellectual Property Rights
Software Integrity Management
Problem Reporting
In Service Software Modifications
Introduction
Migrating to a Software Assurance Standard
Software Assurance Task Matrix
Software Design Acceptance
Commonwealth Oversight
Software Compliance Findings
Software Compliance Finding Technical Specialist
Additional Guidance
Language Selection
Firmware
Operating Systems
Emulation
Software Assurance and Concept Technology Demonstration
Software Tools
Software Compliance Findings
Statement of Work Requirements for Aviation Software
Software Management Plan Template
(Acquirer or CI Manager)
Sample CBD Entries For Software
ADRM (Mar 19)

- Significantly reduced (22+ pages -> 2 pages, no annex)
  - Safety Engineering
  - Software Assurance
  - Software Lifecycle
Safety Engineering Requirements

3.6 **Design Requirement (Essential).** A system safety program must be utilised to determine the worst credible failure condition for all Aviation Software.

3.7 The development of Aviation Software forms part of a larger design effort and is influenced by overarching functional and system requirements. The worst credible failure condition of Aviation Software is dependent on how it influences functional hazards at the system level; this must be established by a system safety program that follows a defined systematic approach for identifying, analysing and evaluating hazards. The TAR prescribed system safety airworthiness design requirements are presented in AAP 7001.054 Section 2 Chapter 2 – System Safety.

3.8 **Design Requirement (Essential).** A software safety program must be utilised in the development of Aviation Software.

3.9 A system safety program developed utilising the TAR prescribed system safety design requirements will require software to be considered in the system context; however in most cases the system safety program will not provide adequate guidelines for conducting the activities required to ensure that the software will execute with an acceptable level of safety. A software safety program provides improved focus on the software aspects of safety analysis, utilising techniques and philosophies designed specifically for the technology that can differ from conventional system safety methodologies.

3.10 A software safety program may be run as part of a holistic system safety program for the aircraft development or modification, or as a dedicated software safety program. The following documents present a robust framework for identifying the relevant tasks, processes, activities, and guidance for inclusion in the software safety program:

Software Assurance Requirements

3.11 **Design Requirement (Essential).** Aviation Software must be assured commensurate with its worst credible failure condition in accordance with the following:

a. RTCA/DO-178C – Software Considerations in Airborne Systems and Equipment Certification; or

b. DEF STAN 00-55 Issue 2 – Requirements for Safety Related Software in Defence Equipment.

3.12 Software assurance is a formal planned and systematic approach to design where all software changes are driven by formal requirements and verified through test, with a range of certification objectives to be satisfied in all phases of the development. The applicability of certification objectives is dependent on the worst credible failure condition of the software, where greater severity requires an increased number of certification objectives. The scope of certification objectives covers the planning, requirements, design development and verification phases of the program, as well as configuration management, quality assurance and certification liaison activities.

Software Lifecycle Management Requirements

3.13 **Software Lifecycle Management Requirement (Recommended).** Aviation Software should be developed and managed utilising a defined systematic software lifecycle process.

3.14 A defined systematic approach to acquiring, developing and supporting aviation software will provide increased support to an aircraft’s initial Type Certification program, as well as ongoing maintenance of the aircraft’s Type Design. While not an essential design requirement, effective management of the software lifecycle ensures that compliance with the Aviation Software design requirements is considered at the planning stage, throughout development, and in-service. Software lifecycle considerations can include supportability, documentation, tools, intellectual property, personnel expertise, interface control, problem reporting and tracking systems, language selection, and hardware requirements.

3.15 Where the Aviation Software design requirements are comprehensively fulfilled, the need to analyse the software lifecycle is reduced, as the formal safety and assurance programs generate the activities and evidence that would otherwise require robust management of the software lifecycle. In managing the Aviation Software lifecycle, consideration should be given to the relevant tasks, processes, activities, and guidance presented in the following:

TAREG, DASR, ADRM Redux

- TAREGs 2.2.11, 3.5.3
  - Split across multiple DASRs
  - SIMS to be incorporated through DAS (21.A.239)
  - Software treated as per any other technology
    - PSAC in CPP (incorporates SCFP?)
    - Software LoI in CPP
    - SOIs 1-4 Reports
- ADRM
  - Reflection of reduction in software-specific regulatory requirements
  - MUCH less prescriptive
  - Where to from here?
Future Direction
Future Direction

WARNING:
The following images and/or content may be disturbing/offensive to some viewers.

Viewer discretion is strongly advised.
Future Direction

Better CUBE?

Future Direction

Overarching Properties (aka Meta Objectives)

- Desired System Behaviour
- Defined Intended Function
- Development
- Implementation

Correctness

Intent

Necessity

OP

OP

Future Direction
Future Direction

Software Safety

Software Assurance

 Derived Req

 Haz Treatment
Sneak Peak - ADRM

AVIATION SOFTWARE DESIGN REQUIREMENTS

6. The application of aviation software design for Defence aircraft achieves two key outcomes:
   a. **A Software Safety system**, integrated with the system safety program. Software hazards are identified, analysed and treated within the context of good systems engineering principles.
   b. A Software Assurance System that provides a framework under which safe software is produced and will perform its intended functions under any foreseeable operating condition over the intended usage life cycle.

7. Authority prescribed software design requirements that support the achievement of each of these outcomes are defined in the following paragraphs.

Software Safety - Identifying Software Hazards

8. Software Safety is an approach that requires the combination of system safety and software assurance disciplines, so that Aviation Software is designed and developed to meet the required safety objectives (commensurate with the software criticality) and functions correctly. Only through the application of both disciplines in concert can the hazards associated with aviation software be appropriately contextualised, analysed, assessed, and treated.

9. System safety covers the application of techniques and philosophies to identify, analyse and treat any condition or event that can cause death, injury, occupational illness, or damage to or loss of equipment or property. Refer to Section 2, Chapter 2 to AAP7001.054 for guidance on system safety.

10. Software safety is a subset of system safety. A major distinction of software safety assessments against other system safety aspects is the absence of likelihood probabilities for each consequence; i.e., while the consequences of software failures can be clearly identified, software risks cannot be characterised in terms of likelihood. Instead, the worst credible failure conditions are assessed utilising methods such as Functional Hazard Assessments for eventual mitigation through methods such as software assurance. The worst credible failure condition of aviation software is dependent on how it influences functional hazards at the system level; this must be established by a system safety program that follows a defined systematic approach for identifying, analysing and evaluating hazards.

11. **Design Requirement (Essential).** A system safety analysis must be utilised to determine the worst credible failure condition for all Aviation Software.
Sneak Peak - ADRM

15. A software safety program should be conducted as part of a holistic system safety program for the aircraft development or modification. The following documents present robust frameworks for identifying the relevant tasks, processes, activities, and guidance for inclusion in the software safety program:


Software Assurance System – An ecosystem of activities

16. The development of aviation software forms only part of the design effort for the entire platform and is influenced by overarching functional and system requirements. It is important to realise that there is not a single method to managing aviation software, but each method describes and implies an ecosystem of activities that must be undertaken. The ecosystem of activities are captured in a number of standards, handbooks and guidance material. As an example, the ecosystem can include ARP4761 as a System Safety standard, ARP4754A for System Development, DO-178C for Software Development and DO-254 for Hardware Development. The Software Development program may then implement manufacturer-tailored coding, design and requirements guidelines.

17. Confidence and evidence that a software product or process satisfies prescribed requirements is achieved through applying appropriate engineering rigour in a planned and systematic approach. The appropriate amount of engineering rigour is determined through the system safety and software safety process described previously. The worst credible failure condition identified is utilised to determine a Development Assurance Level (DAL).

18. Design Requirement (Essential). Aviation Software must satisfy the prescribed safety objectives required for the Design Assurance Level commensurate with its worst credible failure condition in accordance with the following:

   a. EASA AMC 20-115D – “Airborne Software Development Assurance Using EUROCAE ED-12 and RTCA DO-178”; or


UNCLASSIFIED
Sneak Peak - ADRM

AUTHORITY RECOGNISED NAA/MAA APPROACHES TO AVIATION SOFTWARE

23. Section 1 Chapter 3 describes the Authority approach for Airworthiness Codes, which represent prescribed airworthiness requirements from a recognised civil or military Airworthiness Authority. The Airworthiness Codes do not specify a single standard or method to manage the integrity of aviation software, but each describes and implies an ecosystem of activities that must be undertaken. A visual example of the ecosystem provided by civil authorities, in this case for the FAR 25 airworthiness code, is provided in Annex A. Intelligent application of the ecosystem as utilized by the NAA/MAA approach meets the airworthiness design requirements detailed in this chapter and will satisfy the overall software safety outcomes. A mapping between the airworthiness design requirements detailed in this chapter and the ecosystem utilised by EASA/FAA is provided at Annex B and demonstrates how the airworthiness design requirements are addressed by the NAA/MAA conceptual approach. There may be instances, however, where established NAA/MAA approaches do not completely fulfil the software safety outcomes identified in this chapter. This section discusses identified deficiencies in Authority recognised Airworthiness Code approaches to management of aviation software and the activities required to address these deficiencies.

Civil Aviation Software Approach

24. NAA’s including the Federal Aviation Administration (FAA), the European Aviation Safety Agency (EASA) and the Civil Aviation Safety Authority (CASA) recognise RTCA DO-178C and EUROCAE ED-12C, both titled Software Considerations in Airborne Systems and Equipment Certification, as an acceptable means of compliance for aviation software development assurance; and for the satisfaction of the Airworthiness Requirements (generally Federal Aviation Regulation (FAR) parts or Certification Specification (CS) 2x.1301 and 2x.1309). These standards are not intended to be implemented in isolation, and their acceptability is based on integration with other requirements of the relevant Airworthiness Code.

25. Designs certified by an Authority recognised NAA. For new aircraft or modifications to existing aircraft, the extant civil certification demonstrates the adequacy of aviation software for the purposes of Defence aircraft type certification, where the Defence aircraft will operate in the same configuration, role and operating environment (CRE) as that approved by the certifying authority. Application of the civil aviation software standards within the established civil authority ecosystem, will demonstrate compliance with all Authority prescribed aviation software requirements.

26. For NAA certified designs where the Defence aircraft CRE is outside of the scope of the CRE approved by the original certifying NAA. The elements of the design that are outside of the NAA approved CRE must demonstrate compliance with the Authority’s prescribed aviation software design requirements. Where the prescribed aviation software design requirements are not met, additional analysis of impacted hazards will be required.
Other ADRM Chapters?

- Liaising with C4I Command Systems (A8 & A6)
- Mission Planning System
  - Content to remain
  - Aeronautical Data
    - ANSP Regs -> ACPA
      - Certification of AIS-AF to DO-200(B)
- Electronic Flight Bags
  - Content to remain
    - EASA Decision 2014/001/R
    - FAA AC 120-76D or 173
- Cyber security in Airworthiness
  - DO-326A, 355, 356A