



**The Certification of Systems containing Software
Developed using RTCA DO-178B**

**Carolyn Salmon
Clive Lee**

Ref: ASSC/12/0013 Issue 3

June 2006

This page is left intentionally blank

**The Certification of Systems
containing Software Developed using
RTCA DO-178B**

C Salmon
C Lee

ERA Report 2006-0036 Issue 3

ERA Project 7D0134809

Client : MOD

Client Reference : FBG/01189

ERA Report Checked by:

Approved by:



C Hall
Senior Safety Engineer
Safety Engineering

K Moore
Head of Safety Engineering

June 06

Ref: 7D0134809/4010

© Copyright ERA Technology Limited 2006
All Rights Reserved

No part of this document may be copied or otherwise reproduced without the prior written permission of ERA Technology Limited. If received electronically, recipient is permitted to make such copies as are necessary to view the document on a computer system; comply with a reasonable corporate computer data protection and back-up policy and produce one paper copy for personal use.

Distribution list

MoD	(1)
ASSC	(30)
Project File	(1)

DOCUMENT CONTROL

Distribution of this document by the recipient(s) is authorised in accordance with the following commercial restrictive markings:

Commercial-in-confidence : No distribution or disclosure outside of the recipient's organisation is permitted without the prior written permission of ERA Technology Limited.

Distributed-in-confidence : Distribution of the document shall be in accordance with the document distribution list and no further distribution or disclosure shall be allowed without the prior written permission of ERA Technology Limited.

Recipient-in-confidence : ERA Technology Limited distributes this document to the recipient on the condition that no further distribution or disclosure by the recipient shall be allowed.

Where specified the document may only be used in accordance with the 'Purpose of Distribution' notice displayed on the cover page.

For the purpose of these conditions, the recipient's organisation shall not include parent or subsidiary organisations.

Permission to disclose within recipient's organisation does not extend to allowing access to the document via Internet, Intranet or other web-based computer systems.

Commercial restrictive markings are as contained in page header blocks.

If no restrictive markings are shown, the document may be distributed freely in whole, without alteration, subject to Copyright.

ERA Technology Ltd
Cleeve Road
Leatherhead
Surrey KT22 7SA
UK
Tel : +44 (0) 1372 367000
Fax: +44 (0) 1372 367099
E-mail: info@era.co.uk

Read more about ERA Technology on our Internet page at: <http://www.era.co.uk/>

Summary

The Ministry of Defence (MoD) procures many systems containing software which has been previously developed to non-UK or non-military safety standards. The use of previously developed software has a number of advantages, cost reduction being particularly attractive. However, the overall cost of software includes the cost of its acceptance as well as the cost of its development which may be heavily discounted for software already written. The acceptance cost includes all the additional software-related activities required for the system to obtain military certification.

Increase in Prominence of DO-178B for all Airborne Software

The DO-178B¹ guidance document entitled “Software Considerations in Airborne Systems and Equipment Certification” is being widely applied by software development programmes to help the certification of civil avionic systems by the regulator such as the FAA². It has become a “de facto” standard for the software aspects of the certification of these systems for use on civil aircraft.

For US military systems, the 1994 Perry initiative to move towards civilian standards has led to the withdrawal of software standards such as Mil-Std-498 “Software Development and Documentation”. In the absence of other guidance for avionic software, the DO-178B document has gradually assumed more importance as suppliers developing systems for both military and civilian aircraft have incorporated the guidance into their procedures.

In the increasingly global avionics market, many international companies such as Airbus have followed the US lead towards the use of DO-178B. Furthermore, countries which until recently have maintained independent military standards for software safety (such as the UK’s Def Stan 00-55) are also following the US initiative and looking for the private sector to provide and maintain standards.

Software Contribution to System Certification

New UK military airborne applications are now often based on existing systems with a considerable amount of the software being developed using the DO-178B guidelines. The software may have been assessed as part of the certification of the existing system in either a military or civil context.

To enable efficient certification of new UK military applications, an understanding of the impact and scope of the use of DO-178B is essential. There are two main issues:

1. the way in which DO-178B was actually applied in the existing system
2. the other safety activities undertaken for certification that complement DO-178B guidance.

¹ European version is referenced as ED-12B, DO-178B/ED-12B was produced by RTCA/EUROCAE jointly.

² In Europe by the EASA (European Safety Aviation Agency) supported by national agencies such as the CAA.

There is a current belief within the MoD that DO-178B is insufficient for the UK military safety requirements and that significant additional activities need to be undertaken before certification can be gained, reducing the significant benefits, particularly in terms of time and cost, presented by procuring systems which are already available.

Experience of Procurement

From the investigations undertaken, some past procurement programmes had found that the evidence presented was insufficient (for UK military purposes) to demonstrate the required integrity level of previously developed software. In these cases additional assessment activities had been undertaken, often at considerable cost, to gain the level of confidence required.

For example, the military standard for software safety at the time, Def-Stan 00-55, advocated the use of Static Code Analysis (SCA). SCA covers a range of techniques from simple inspection and type checking to special tools that enable the compliance of the code to be checked against requirements. At the highest levels of integrity, the standard demanded very rigorous techniques which required even more effort when undertaken retrospectively.

DO-178B does not mention SCA directly but requires verification by review and analysis to complement testing. The requirements for SCA can be included in the Software Verification Plan agreed by the Designated Engineering Representative on behalf of the regulator.

Static Code Analysis

The requirement for more rigorous SCA does not necessarily mean that the UK military approach to safety is more rigorous in general. For example, the “system safety approach”³ of the US military intentionally directs more effort at system safety analyses rather than requiring the correctness of the code to be demonstrated by SCA. This approach is based on the observation that more and more accidents in complex systems occur due to misunderstood or unexpected system and human interactions; the individual systems actually behaving according to their specifications.

IPT Feedback

From discussions with the IPTs, it is apparent that software-related problems in previous programmes are not predominantly related to deficiencies directly in the DO-178B guidance, but arise through other factors. Important issues are lack of management of contractual requirements, lack of visibility and access to subcontractor’s documentation, and lack of confidence in the emerging (at that time) software engineering technology.

³ Leveson and others

Since these early procurement programmes, software engineering has evolved and the management of contractual arrangements has significantly improved. However, there remains the perception that the DO-178B guidance is not sufficient for UK military applications, and that costly retrospective SCA is essential before a Release To Service (RTS) recommendation will be forthcoming. Costs of \$13 million have been estimated for the application of retrospective SCA per aircraft type. For aircraft being purchased “off-the-shelf”, such as the C-17 Globemaster, such huge additional costs in order to gain certification are being questioned, and there is a need to reconsider whether the use of retrospective SCA is the most cost effective solution.

Key Issues of Study

The objective of the study programme reported in this document is to examine the problems that the UK MoD faces in trying to determine whether or not to accept (as sufficiently safe) software developed to DO-178B. In particular, to establish, from both published material and from discussion with the IPTs whether the perception that DO-178B is insufficient is justified, and to propose alternative approaches to retrospective SCA for gaining certification of systems which use software that has been previously developed to DO-178B (with a view to formulating guidance on the subject in future work).

From the investigations undertaken a number of key issues have become apparent. Firstly, it is worth emphasising that there has been variation in the rigour of application of DO-178B guidance, a key factor being the competence of the engineers which hugely impacts on the quality of software developed. For MoD purposes, it is necessary to establish what has been done, maybe over and above the objectives of DO-178B and its accompanying certification framework, that has led to the development of good software and how this can be used to ease the burden of re-certification of that software for a different application. Both Airbus and Boeing are considered to be capable of developing good quality software, and it is apparent that such organisations have developed internal procedures which take the requirements of standards and guidance, and have enhanced them to ensure a complete process for software and system development.

Even though there have been improvements in software development in general, there is still much disparity between the IPTs views on whether DO-178B is good enough for current software development programmes. Some IPTs are prepared to accept the existing certification with minimal review, while others will take very little account of this and start developing their own evidence to gain certification from scratch. There is some convergence in the IPTs approaches by concentrating effort on military deltas which is both pragmatic and appropriate, but there may be scope for reducing the effort further if there was a greater confidence in software developed to DO-178B.

It is also noted that the role of how software has been certified in a US civil application does not seem to be widely understood, and there is much benefit in ensuring that there is a wider appreciation on the context of software being certified as part of a system and not standalone. The IPTs need guidance on how to take the existing certification evidence and the approaches that could be followed in order to

gain certification of their product in a UK military application. Currently, each IPT is developing its own approach.

This report concludes by providing alternative approaches to gaining certification of systems using software developed to DO-178B. The approaches are only provided in outline, as further funding is required to develop the approaches into guidance material for wider application by the IPTs. The approaches are focused on gaining certification for system using existing software, rather than programmes which are developing bespoke software. This existing software is considered from two different perspectives, one for software developed for a specific application with the system application being certified and for software developed for a generic application which may have not yet been used in a certified application.

The IPTs interviewed identified such guidance as necessary to provide the confidence that the approach that they are adopting is a suitable alternative to retrospective SCA. This guidance material will provide explanation on how software developed to DO-178B is certified in a system context. Without this understanding, the awareness of the system context is lost, and there is likely to be a lack of appreciation of what information could be available to aid in the development of a safety argument. For example, there is a wide concern amongst the IPTs and industry as a whole that there is a lack of a safety management system in DO-178B. This is intentional, as safety is addressed at the system level. However, the awareness of this concept is not extensive and the perception remains that safety is not sufficiently addressed.

As with any complex activity, guidance material which provides a consistent approach will always be beneficial. IPTs will be given greater confidence that the approaches they are adopting will be acceptable to independent assessors and advisors. Concerns over the rejection of programmes late on in the lifecycle and the need for extensive rework will be lessened. The guidance material will also lead to a better understanding of the approaches and more consistency in application amongst the IPTs, which in turn will result in improved efficiency, in terms of the burden of both time and cost, in the certification process.

In addition, the assessors and advisors will be more efficient as they will be better prepared for the approaches being followed. They will be able to focus on the key issues instead of addressing whether the overall approach is sufficient. This focus should lead to a greater confidence in the safety argument being presented to the Release to Service Authority, who in turn will be better prepared and more readily able to understand the submission for certification if a consistent approach is adopted across the IPTs.

In conclusion, there is currently a wide disparity across the IPTs in the approaches being taken to gain certification. From a purely financial perspective, a single approach which reduces the currently high costs of certification would be more desirable. However, no such guidance exists, and the IPTs have expressed a desire for guidance to be developed, focussing on the issues with regard to the certification of system using software developed to DO-178B.

This page is intentionally left blank

Contents

1	Introduction	15
1.1	Background	15
1.2	Objectives and Target Audience	15
1.3	Guidance for Systems containing previously Developed Software	15
1.4	Scope of Study	16
1.5	Study Approach	16
2	DO-178B	18
2.1	Introduction	18
2.2	Reliability	18
2.3	Safety	18
2.4	DO-178B Status	19
2.5	Certification of Systems with Software Developed to DO-178B	20
	2.5.1 Certification overview	20
	2.5.2 System safety assessment	21
	2.5.3 Applicability of justification in new system context	21
2.6	Comparison of DO-178B with Other Standards	22
	2.6.1 Omission of safety management system in DO-178B	22
	2.6.2 Formal methods and static code analysis	24
2.7	Industry View	24
	2.7.1 US civil market	24
	2.7.2 UK military market	27
2.8	Initial IPT Feedback	34

3	Use of Static Code Analysis	38
3.1	Introduction	38
3.2	Past Experience	38
4	Conclusions	41
4.1	Variation in the Quality of Application of DO-178B for Certification	41
4.2	Interpretation and Extension of DO-178B	41
4.3	Some issues for High Integrity Software	41
4.4	Quality and Perception of DO-178B Software	42
4.5	Content of Certification of Software Systems	43
5	Proposed Approaches to Certification	44
5.1	Introduction	44
5.2	Common Standards	44
5.3	New Programmes	45
5.3.1	Mapping Civil to Military Standards	45
5.4	Existing Developments to DO-178B	47
5.4.1	Certified software	47
5.4.2	Non-certified software	49
5.5	Existing Development to Other Standards	50
6	References	51
Appendix A	Previous Research and Publications	55
A.1	SSAC (1998) – US Industry concern over DO-178B	56
A.2	Boeing (1998) – Comparison of DO-178B with MIL STD 498	58
A.3	YSE (1999) - Comparison of DO-178B with 00-55	59

A.3.1	Scope	59
A.3.2	Comparison of RTCA DO-178B against Def Stan 00-55	59
A.3.3	Comparison of Def Stan 00-55 against DO-178B	65
A.4	DERA (1999) - A comparison of Avionics Standards	71
A.5	Adelard (2001) – COTS and Safety Related Applications	72
A.6	ERA (2002) - Comparison of Def Stan 00-56 and SAE ARP 4761/4754	72
Appendix B	DO-178C	74

Abbreviations List

AC	Advisory Circular
ADRP	Airworthiness Design Requirements and Procedures
AD/ADRP	Assistant Director/ADRP
ALARP	As Low As Reasonably Practicable
AMC	Acceptable Means of Compliance
ARP	Aerospace Recommended Practice
ASSC	Avionics System Standardisation Committee
CAA	Civil Aviation Authority
CQC	Certification and Qualification Committee
CQO	Certification and Qualification Organisation
CS	Certification Specification
CSS	MCSP (q.v.) Software Standard
CTW	Crisis, Tension and War
DAL	Design Assurance Level
DAL	Development Assurance Level
DCMA	Defense Contract Management Agency
DER	Designated Engineering Representatives
DERA	Defence Evaluation and Research Agency
EAP	Evidence Assurance Plan
EASA	European Aviation Safety Agency
FAA	Federal Aviation Administration
FADEC	Full Authority Digital Electronic Control
FAR	Federal Aviation Regulation
FSTA	Future Strategic Tanker Aircraft
GATM	Global Air Traffic Management
INTA	Instituto Nacional De Technica Aeroespacial
IPT	Integrated Project Team
ISA	Independent Safety Auditor
loc	Lines of code,+ variants such as sloc,kloc, skloc source thousand loc
MALPAS	Malvern Program Analysis Suite
MCSP	Merlin Capability Sustainment Plus /Programme
MoD	Ministry of Defence
OCCAR	Organisation Conjointe de Coopération en matière d'Armement
OSHA	Operational and Support Hazard Analysis

QQ	QinetiQ
QQ(BD)	QinetiQ (Boscombe Down)
RTCA	Radio Technical Commission for Aeronautics
RTS	Release to Service
SAS	Software Accomplishment Summary
SCA	Static Code Analysis
SCP	Software Certification Plan
SCWG	Special Committee Working Group
SIL	Safety Integrity Level
SOUP	Software of Uncertain Pedigree
SSAC	Streamlining Software Aspects of Certification
USG	US Government
YSE	York Software Engineering

1 Introduction

1.1 Background

The Ministry of Defence (MoD) procures many systems containing software which has been previously developed to non-UK or, increasingly, non-military safety standards. The use of previously developed software has a number of advantages, cost reduction being particularly attractive. However, the overall software cost includes the cost of its acceptance, that is the cost of the activities required for the system to obtain military certification, as well as the cost of its development which may be heavily discounted for previously developed software.

For a number of these systems, the safety evidence relating to the software has been found to be inadequate for MoD requirements during independent safety assessment. In particular, the development and safety standards employed were considered to be inadequate, or were applied inappropriately, and did not provide the evidence required. In these cases, it has been difficult for the IPT concerned to determine a practical way forward to obtain further safety evidence retrospectively, often after the development team has disbanded.

For several high integrity systems, costly post development assessment activities, such as Static Code Analysis (SCA) of all the software, have been undertaken to demonstrate safety integrity and enable UK military certification. As an extreme example, the cost of MALPAS compliance analysis (a form of SCA) can be several times the original cost of development of the software.

1.2 Objectives and Target Audience

The objectives of the study reported here were to:

- Examine the problems encountered by the MoD in determining whether or not to accept (as sufficiently safe) software developed to RTCA/DO-178B
- Consider potential alternatives to retrospective SCA for MoD acceptance of DO-178B software.

Although it is not within the scope of this report to meet it, an overarching objective is to subsequently provide guidance to the MoD on the acceptance of software developed to RTCA/DO-178B.

The primary target audience for this report is persons within the MoD with responsibilities for safety-related avionics software.

1.3 Guidance for Systems containing previously Developed Software

The demonstration of the safety of systems containing previously developed software has been thoroughly addressed over the past decade with some progress, but little authoritative guidance is available. In particular, the IPTs, system integrators, software developers and other stakeholders in these systems are seeking guidance and consensus in the following areas:

1. determining the “credit” that can be claimed from the software evidence obtained for the certification of the previous systems in which it has been used; this “credit” may also extend to assurance gained from a significant history of use
2. identifying the extra evidence required for UK military certification (additional to the evidence provided for the certification of other systems using the software)
3. determining a cost effective way of obtaining the extra evidence.

Although there have been some detailed project reviews, there has been no definitive study which has identified the common shortfalls in the safety evidence relating to previously developed software.

1.4 Scope of Study

The study reported here started with a proposal to the ASSC to undertake a formal comparison of all relevant software and safety standards relating to the development and acceptance of systems using previously developed software. The intention was to identify their differences and similarities and determine the potential for cross acceptance for military certification. The aim was to see if the burden of achieving compliance of previously certified software to UK military standards could be reduced.

The proposal was presented to an ASSC workshop attended by IPT and industry representatives. During the meeting it was agreed that the scope of the study should be restricted to software developed to the civil document RTCA DO-178B “Software Considerations in Airborne Systems and Equipment Certification” (Ref. 1). The study was also directed to consider the further safety activities that may need to be undertaken to achieve the requirements of Issue 3 of Def Stan 00-56 (Ref. 2) for a UK military application.

Finally, the workshop also requested the study to propose some alternative approaches to full retrospective Static Code Analysis (SCA) in order to gain certification where the appropriate evidence could not be obtained. The study should address the identification of the reasons that SCA had been employed as a basis for proposing alternatives.

1.5 Study Approach

The main approach to the study was to consider the following questions:

1. Why is DO-178B not considered sufficient? A review of the differences identified between DO-178B and comparable military standards as documented in existing research papers was undertaken to establish where the differences lie and the implication of these differences. In addition, industry opinion was sought to establish where current thinking lies, and whether this is consistent with the issues identified in the formal comparisons. This is covered in Section 2 of the report. For background information, Section 2 also provides an introduction

to the scope and status of DO-178B, and how certification of systems using software developed to DO-178B is currently achieved.

2. Why is the retrospective SCA approach advocated? The objective of answering this question was to establish who is recommending that SCA be undertaken and why this approach is considered to be appropriate to address perceived weaknesses in the DO-178B approach. In addition, the cost and effectiveness of undertaking retrospective SCA was investigated. This is covered in Section 3 of the report.
3. What other approaches could be undertaken? Proposed alternative approaches are outlined. Further work is required to establish fully the approach to military certification for systems with software certified to DO-178B. This is covered in Section 5 of the report.

For background information, previous research undertaken or that is currently ongoing was reviewed. Reference to this research is provided throughout the report, with detailed information on previous research provided in Appendix A.

In addition, it is recognised that DO-178B is currently being re-written. A brief summary of the proposed changes to the guidance document is provided in Appendix B.

2 DO-178B

2.1 Introduction

DO-178B⁴ “Software Considerations in Airborne Systems and Equipment Certification” published by the Radio Technical Commission for Aeronautics (RTCA) is a guidance document to address the particular issues concerned with software that arise during the certification of a system.

It is important to appreciate that the document is produced in the context of the whole infrastructure connected with the civil certification of airborne systems and equipment. A lot of comments about “omissions” from the document are not relevant to civil systems as the issues are addressed elsewhere in the regulatory and guidance documentation for the certification process.

2.2 Reliability

The major part of DO-178B provides guidance on the activities to gain assurance that the software works correctly and reliably. Indeed, a common observation on the document (often made critically) is that it addresses software reliability rather than software safety; that is, it does not fully address the activities concerned with analysing the safety-related failures of the software.

If the document is considered in isolation, there is undoubtedly an emphasis on the demonstration of reliability of the software by functional and robustness testing. There is a requirement for the testing to be complemented by review and analysis, but little detailed guidance is provided.

2.3 Safety

The DO-178B document should be seen as software guidance to support the demonstration of safety required for system certification, that is to support the system safety assessment.

The consequences of component failures, including software failures, are assumed to be analysed by the system safety assessment process to determine their possible contribution to system and aircraft failure conditions. This assessment is not within the scope of DO-178B but is dealt with in detail in other parts of the certification process.

An important output of the system safety assessment is the designation is Software Level based upon the potential contribution of the anomalous behaviour of the software to an aircraft failure condition. The Software Level indicates the techniques, rigour and effort required to meet the DO-178B objectives.

⁴ DO-178B is recognised by European Organisation for Civil Aviation Equipment (EUROCAE), where it is known as ED-12B.

DO-178B does provide some guidance on the software architectures and software development techniques that have been used to achieve system safety goals, for example:

1. “Partitioning” to isolate faults
2. “Multiple-Version Dissimilar Software” as a fault tolerance technique
3. “Safety Monitoring” for protection against specific failure conditions.

A potential weakness of DO-178B is that there is little guidance concerning the safety analyses that should be applied at each level of the software design and integration. For systems of the highest integrity, these could include software FMECAs, software FTAs or related analyses on design artefacts and code. In practice, these analyses are driven by the system safety process and planned and agreed with the DER, rather than being addressed through adherence to DO-178B.

2.4 DO-178B Status

For the civil certification of systems, DO-178B/ED-12B has become the “de facto” standard for obtaining approval of software to meet the requirements of the Federal Aviation Regulations (FARs), or the equivalent Certification Specifications (CSs) of the European Aviation Safety Agency (EASA), represented by the Civil Aviation Agency (CAA) in the UK. An important step in achieving this status is given by the FAA Advisory Circular #20-115B (Ref. 3) which specifies the use of DO-178B as an acceptable means of compliance with the software aspects of the FAR Clause 25.1309, “Systems, Equipment and Installations”.

Furthermore, under the Global Aviation Traffic Management (GATM) agreement, all commercial airborne systems have to comply with FAA/EASA regulations for avionics and require certification. Consequently, all airborne military and space systems which will be flown in civil airspace must comply with FAR/CS 25.1309. In practice, this means that the software guidance of DO-178B will also be used to support the system certification process of military and space systems.

These sweeping requirements mean that much of the previously developed software to be utilised in UK military avionics applications is increasingly likely to have been developed to the DO-178B guidance and the evidence may have contributed to FAA/EASA certification. Therefore, if the MoD wants the option to procure equipment from as many suppliers as possible, it needs to embrace the DO-178B guidance and fully understand the civil certification process.

The following sections of the report identify:

1. The route to certification for software developed in accordance with DO-178B; the purpose of this section is to explain how software is certified by the FAA in the context of its system application and not, as is sometimes perceived, as a stand-alone software component (2.5).

2. How DO-178B differs to similar standards based on findings of existing research; the purpose of this section is to review the comparisons, which have previously been made, between DO-178B and other similar standards. These published comparisons are discussed and an opinion is provided on the validity of any criticisms raised in the context of the information provided with respect to certification (2.6).
3. Finally, some of the criticisms of DO-178B based on industry opinion, rather than research, are provided. The purpose of this is to try to establish, from experience, what are some of the difficulties in developing software to DO-178B and subsequently gaining certification for the use of that software in civil or military applications(2.7).

2.5 Certification of Systems with Software Developed to DO-178B

The use of previously developed and certified software has the potential for significant cost reduction in the procurement of systems by the MoD. However, a full understanding of the process and context in which the software has been certified is essential for a safe and efficient procurement that eliminates unnecessary duplication of assessment effort.

2.5.1 Certification overview

To achieve “certification”, a system must comply with all applicable requirements of the FAA/EASA regulations. For an application on a large civilian transport aircraft, subchapter FAR/CS 25.1309⁵ (Ref. 4) is particularly important as it addresses the requirement for the safe design of the system and the demonstration of its safety.

For subchapter FAR 25.1309, an Acceptable Means of Compliance (AMC) has been provided in an accompanying Advisory Circular (AC) 25.1309⁶ (Ref. 5) which details some methods and guidelines to design a safe system and demonstrate its safety. In addition to this “System Design and Analysis” AMC, further software guidance is provided by DO-178B.

The system will be “certified” only when compliance with the regulations has been demonstrated to the regulator or his representative. For the example above, this means the system must be demonstrated to be safe by system safety assessment by following AMC 25.1309 (or equivalent means) with the software aspects addressed by DO-178B (or equivalent means).

⁵ The FAR and CS 25.1309 are textually identical.

⁶ In Europe, the AMCs are labelled appropriately (e.g. AMC 25.1309) and stored together as Book 2 of the CSs.

2.5.2 System safety assessment

The system safety assessment is an “integral” or continuous process undertaken throughout the development of the system. In particular, hazard identification and analysis is conducted at various levels as the design is consolidated e.g. at the preliminary stage, at system level and possibly down to subsystem and software levels.

The principle of continued safety assessment on the detailed design is also applicable to software. The continued safety assessment of software depends on a knowledge of the hazards identified for that particular system. For example, the hazard analysis of the software will address such questions as “How can the failure of the software lead to this hazard?” or “How can we ensure that the failure of this software does not lead to any of these hazards?” or “Can the failure of this software lead to a hazard which has not been identified/considered?”

2.5.3 Applicability of justification in new system context

For the use of previously developed software, the question is how much credit can be claimed from the existing justification or, more practically, how much of the evidence and argument is available and applicable in the new context. A safety justification must be made that any anomalous behaviour of the software cannot cause or contribute to the hazards associated with the new system context.

For the most part, the justification will rest on the evidence and arguments such as testing and the application of the development process that were used for the original application, and whether this is made available. Nevertheless, the operational profile of the software and the specific hazards of the system in the new context must be considered. The justification may require the generation of new or more detailed evidence, for example from further testing, analysis and additional auditing.

Consequently, there will always be some effort required to justify the use of previously developed software in a safety related application. For example, even in the simplest case of non safety-related software, a convincing argument must be made that the software is indeed not safety-related; that it cannot cause or contribute to the hazards associated with the system.

The civil aviation regulator does not accept software at any Software Level (including Level E) in a different system without reassessment. The FAA has produced an Advisory Circular AC 20-148, “Reusable Software Components”, December 2004 (Ref. 6) to provide an acceptable means of compliance for reusable software component (RSC) developers, integrators and applicants. The same principle applies to previously developed software used in military aviation applications.

Some important questions for the MoD are: “What are the special considerations for the acceptance of systems containing software?” and “How can the relevant evidence acquired and presented for civil aircraft certification be identified, accessed and incorporated in a safety justification?”.

2.6 Comparison of DO-178B with Other Standards

Previous research has been undertaken comparing DO-178B with other relevant standards. It is worth reiterating that DO-178B is guidance for the software aspects to support the certification process in which the safety aspects are primarily addressed at system level. The conclusions drawn from a one to one comparison of DO-178B with another document are limited by this context.

In retrospect, the set of requirements and guidance for civil certification such as FAR/CS 25.1309, AMC 25.1309, ARP 4754, ARP 4761, DO-178B, DO-254, DO-160E should be compared with the set for military avionics such as JSP 550 series, JSP 553 in particular, Def Stan 00-56 and recently withdrawn Def Stan 00-54, 00-55, 00-58.

In particular, a study was undertaken in 1999 to compare, clause-by-clause, DO-178B, Def-Stan 00-55 “Requirements for Safety Related Software in Defence Equipment” (Ref. 8) and IEC 61508 (Ref. 9). Although Def Stan 00-55 has been withdrawn or is technically “obsolescent”, it is worth revisiting the results of the study to form a baseline to take the current programme of work forward.

The research identified, amongst others things, two key differences between DO-178B and Def Stan 00-55, in particular:

1. DO-178B does not address safety management or risk assessment. As such there are no requirements for preparation of a Safety Case, identification of safety requirements, undertaking safety audits, maintaining safety records and other activities typically encountered in a Safety Management System.
2. Def Stan 00-55 places emphasis on formal methods and static analysis. Since Def Stan 00-55 is focused on safety critical software (i.e. software classified with the highest software integrity level (SIL) of 4) there is an emphasis on the process for development for software of the highest integrity. In comparison, formal methods and static analysis are only referred to in passing in DO-178B.

Full details on the results of the clause-by-clause comparison and the omissions identified are provided in Appendix A.3 of this report. Key issues are briefly discussed below.

2.6.1 Omission of safety management system in DO-178B

The omission of a safety management system from DO-178B, at first sight, appears to be fundamental in considering whether the software development is likely to be sufficient for acceptance in a UK MoD application. Without it, an argument for the safety of the system is difficult to construct.

However, the context in which DO-178B is applied (and equipment certified as developed in accordance with it) should be considered. DO-178B refers to the “system safety assessment” process (see Section 2.2) as part of the system lifecycle which is outside the scope of the document. At the

time of publication of DO-178B in 1992, there was guidance available on safety assessment in the advisory circular AC 25.1309.1A (Ref. 5) in support of the FAA regulation FAR 25.1309 (Ref. 4).

This guidance (first published in 1988) was very limited in its context and did not provide comprehensive guidance to aid the developer in the conduct of safety assessment activities. Additional supplementary guidance was published in 1996 in the form of ARP 4754 “Certification Considerations for Highly Integrated or Complex Aircraft Systems” (Ref. 10) and its supporting guidance document ARP 4761 “Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment” (Ref. 11). However, these ARPs are not mandatory. Whilst there are many organisations, such as Airbus and Boeing, which appreciate the additional benefit of these recommended practices and use them consistently, there are many developers of civil avionics systems who have little or no experience in applying them. This situation could be improved by making the ARPs an acceptable means of compliance, however an advisory circular stating such has yet to be issued.

Although DO-178B does not cover a safety assessment process, software developed to DO-178B should always be certified as part of a system and not as a standalone component (see Section 2.5). A system safety assessment should be carried out at the system level. All US civil avionic equipment is subject to FAA regulation which requires that a safety assessment be undertaken to demonstrate that the system and its components are sufficiently robust. FAR 25.1309 requires that:

“The airplane systems and associated components [...] must be designed so that:

- (1) The occurrence of any failure condition which would prevent continued safe flight and landing of the aircraft is to be extremely improbable, and
- (2) The occurrence of any other failure condition which would reduce the capability of the airplane or the ability of the crew to cope with adverse operating conditions is improbable.”

Therefore, stating simply that DO-178B does not address safety management is not necessarily representative. In order to gain certification, compliance with FAA regulations will have to be demonstrated, and therefore a safety management system will have been implemented at some level.

DO-178B does include preparation of a Software Accomplishment Summary (SAS), which should provide some of the evidence of software assurance, but the focus of the SAS is demonstrating that the software development complied with relevant plans and DO-178B. Since the software developer may not be responsible for safety management, the SAS is likely to be limited in its scope to the software development process rather than the safety assessment.

It should also be noted that DO-178B explicitly excludes the requirement to show failure rates for software (Section 12.3.4 of the guidance) since it argues that “currently available methods do not provide results in which confidence can be placed to the level required for this purpose”. DO-178B

does refer to “derived requirements” which could include safety requirements. The guidance requires that these derived requirements are “provided to the system safety assessment process”. DO-178B does not provide guidance on how a system safety assessment process should use these derived requirements. However, guidance is available elsewhere such as in the ARPs (Refs. 10 and 11) which include analysis techniques such as the application of a risk assessment to justify that safety requirements have been met.

2.6.2 Formal methods and static code analysis

The original study undertaken in 1999 and reported (Ref. 5) compared DO-178B against Def Stan 00-55 and the relevant sections of Issue 2 of Def Stan 00-56. Since then Def-Stan 00-55 was withdrawn (April 2005), and 00-56 Issue 3 (Ref. 2) has been published (17 December 2004). Although Issue 3 of Def Stan 00-56 now includes a few of the requirements of Def Stan 00-55, a move towards a more goal-based approach has resulted in the loss of many of the detailed process based requirements that appeared in the former software standard. Thus, many of the comparisons made in the original study no longer apply. In particular, activities such as SCA and formal methods no longer warrant discussion in Issue 3 of Def Stan 00-56. The focus of the standard is now very much on the safety assurance programme rather than the detail of the activities typically undertaken for a software development.

In principle, the lack of detailed prescription could make conformance to aspects of Def Stan 00-56 Issue 3 for DO-178B “compliant” software less onerous. However, this observation does not answer the question as to whether adherence to DO-178B is likely to be sufficient for a UK military market. This issue is re-considered and addressed in more detail later in the paper.

2.7 Industry View

2.7.1 US civil market

In 1998, some concerns over the burden of certification of software developed to DO-178B were being raised in the USA. In an attempt to address some of these concerns the FAA introduced the Streamlining Safety Aspects of Certification (SSAC) programme. The program focused on addressing the difficulties associated with developing high integrity software and achieving certification. A public workshop was held and an industry survey was undertaken in 1998 with wide representation.

The results of the activities are published in the following reports:

- Streamlining Software Aspects of Certification: Technical Team Report of the First Industry Workshop (Ref. 14)
- Streamlining Software Aspects of Certification: Report on the SSAC Survey (Ref. 15).

The programme identified that there were many aspects of DO-178B that were inadequate and ambiguous. One key issue identified was that “DO-178B guidance should address how implied requirements that affect safety should be addressed”. In addition, the benefits of some of the activities within DO-178B were queried, and it was felt that DO-178B “inadequately addresses the effect of software on the safety of the overall system.”

A detailed questionnaire on some of these aspects was compiled and views from industry sought. In total, 292 responses were assessed. With regard to the safety issue one of the questions raised was whether the respondents had “Ever worked on a system developed in compliance with DO-178B in which a software-related system error resulted in a service bulletin⁷ or Airworthiness Directive⁸”. 28% of those questioned responded “yes”, and identified the source of these errors mainly from requirements (47%), design (32%) and coding (35%).

This finding highlights two particularly pertinent issues. Firstly, the importance of defining appropriate requirements. There is a lot of evidence, from experience of developing software, that most software problems are at the requirements capture and definition level. As already stated, DO-178B does not include specific objectives for formal definition of requirements, whereas Def Stan 00-55 did. DO-178B does require that software requirements are defined, complied with and are traceable to system requirements, and are accurate and consistent. However, the use of formal specification methods and notations is not specifically required by DO-178B.

Secondly, in the US a number of safety issues have been identified “in-service” relating to software claimed to have been developed to DO-178B. This could be interpreted as a weakness in the guidance and/or the way it is being applied. Alternatively, it could indicate that the “in-service” Safety Management System is efficient; indeed, the safety record for commercial aircraft is very high with a very low number of accidents. Nevertheless, there is a perception that there should be fewer safety-related software problems detected “in-service”.

The study was undertaken in 1998, when experience with DO-178B was still relatively new and the ARPs had only been available for 2 years, so many of the findings would be based on software developed to DO-178B without the benefit of the system assessment guidance provided by the ARPs. It is assumed that the same survey undertaken today would generate improved results (i.e. a lower occurrence of safety bulletins or airworthiness directives) due to the improvements in the safety assessment process and a better understanding of the requirements of DO-178B in the context of the system safety assessment process.

⁷ Service Bulletins address items identified through service experience that affect safety of flight.

⁸ Airworthiness Directives are issued by the FAA as notification of a known safety issue. These directives may be voluntary, mandatory or emergency.

It is unfortunate that funding for the SSAC programme of work was withdrawn. In particular, the relationship between DO-178B and the safety assessment process still presents a significant area of concern for software development programmes. It is one of the areas that is intended will be addressed through the updating and development of DO-178B to DO-178C (see Appendix B).

However, since the time of the SSAC programme, HighRely in the US have published a number of whitepapers on issues relating to DO-178B based on their extensive experience in the field. Two whitepapers of particular interest relate to certification (Ref. 16) and Costs versus Benefits (Ref. 17). These whitepapers indicate the significant benefits of utilising DO-178B which include:

- Greater supplier visibility
- Greater upfront requirements clarity
- Fewer coding iterations
- Greater consistency within software
- Fewer defects found during integration
- Improved hardware integration
- Fewer in-the-field defects
- Improved re-usability
- Decreased single-point errors introduced by individuals
- Improved management awareness
- Improved completion schedule.

The papers recognise that there is a cost implication associated with the application of DO-178B, but it is contested that this is not significant for Level D software, since the processes being adopted should be considered normal industry practice. The papers suggest that cost increase for development of Level C software over Level D software is about 30%, but then further increases for Level B and Level A software are an additional 15% and 20% respectively.

The whitepaper on Military Avionics Certification makes an interesting observation on the use of DO-178B in US military applications. It notes that, in the main, the FAA is not formally involved in military projects and therefore formal certification is not possible. Instead, for these applications, the military will self-certify (although they may seek advice from the FAA just as the CAA is sometimes consulted by the UK Military).

This approach could mean that certain requirements are interpreted differently for these military applications. The potential dangers of self-certification are well known and have been addressed by both the US and UK Military in slightly different ways but it is recognised that software does pose problems of a different order. This observation, again, emphasises the need to fully understand the context in which any system utilising software developed to DO-178B has been certified, since even though the main principles of DO-178B may have been followed some requirements may be interpreted differently. For any aircraft that is intended for operation in civil airspace, FAA

certification will be required, so there may only be a restricted subset of military applications to which this observation may apply.

2.7.2 UK military market

As part of this study, various MoD IPTs were interviewed to determine their views on the adequacy of DO-178B and to identify the processes for certification that have been adopted or proposed. The IPTs were very accommodating and helpful, and have provided a valuable contribution to the study.

The following IPTs were interviewed in November 2005:

1. Merlin Capability Sustainment Programme (MCSP): Jack Kebbell and John Stanley
2. Future Strategic Tanker Aircraft (FSTA): Tim Green
3. A400M: Carl Garvie and Ron Wilkins
4. Hercules: Robert Hull and Ian Rillie
5. Sentry: John Waterhouse
6. Chinook Mk 3: Colin Gale
7. C-17 Globemaster: Andrew Lever
8. Future Lynx: Paul James.

The approach to certification adopted by each IPT is discussed in sections 2.7.2.2 to 2.7.2.9. Preceding these sections, general comments and discussions on the adequacy of DO-178B are also provided.

2.7.2.1 General comments

The IPTs involved in the discussions fell in two groups; those responsible for rotary wing aircraft and those responsible for fixed winged aircraft based on existing civil aircraft. There were limited discussions with fast jet IPTs (such as Harrier or Tornado) since they did not appear to use software developed to DO-178B, which was the primary focus of the study. None of the projects are novel, with all based on aircraft that have been operational for many years, and certainly in many cases predating the introduction of DO-178B.

In discussing past projects, many of the IPT's noted that their problems rested with contractual arrangements, rather than a direct deficiency in DO-178B. Problems encountered repeatedly included:

1. Lack of access to subcontractors' documentation and source code preventing the development of an appropriate safety argument

2. Contractual requirements (such as developing software in accordance with current best practice) not being flowed down to subcontractors
3. Lack of experience of software developers for safety applications (such developments were novel at the time and much of today's best practice advice was not available).

These factors were key in the lack of confidence in the software development activities undertaken, and significantly impacted the IPTs ability to prepare appropriate safety arguments. However, since these original procurement programmes, software engineering techniques have evolved significantly and as a result, much greater confidence is being placed in suppliers' ability to develop robust software in accordance with best practice. In addition, prime contractors are becoming more familiar with stringent contractual arrangements and have made improvements to their procedures and processes to address previously experienced difficulties.

Although many of the IPTs acknowledged that SCA may be essential to gain certification, many of the current programmes of work avoided the use of SCA as far as possible. There appears to be a widely held view that, although retrospective SCA is costly, time consuming and provides little tangible benefit, it is necessary in order to gain a Release To Service (RTS) Recommendation from the safety assessors, such as QinetiQ (Boscombe Down) (QQ(BD)). The source of such hypotheses is unclear as in reality, many of the approaches that do not utilise retrospective SCA are also supported by QQ(BD).

Already a more pragmatic approach to certification is being adopted by the IPTs with the appropriate support of the safety assessors. There is some commonality in the approaches, being proposed but there is also some divergence. For example, many IPTs are now considering more pragmatic approaches to certification which entails a review of the existing evidence and addressing military deltas. The main exception is the Hercules project, which is still following the approach which relies heavily on retrospective SCA to provide the level of assurance of safety needed. Some safety programmes are well developed with appropriate documentation in place, while others are still to be fully developed. All IPTs felt they would benefit from a single guidance document describing alternative appropriate approaches which are widely agreed as pragmatic and effective.

The following sections provide details of the discussions held with the IPTs.

2.7.2.2 Merlin Capability Sustainment Project (MCSP)

The original contract for Merlin was initiated in the 1980s with much of the software being developed to DO-178A. At that time the ARPs were not published. The current programme, the Merlin Capability Sustainment Project (MCSP), is to undertake a technology refresh of the avionics, with software being developed in accordance with the requirements of Def Stan 00-56, Issue 3. Interestingly, the contract indicates that Def Stan 00-55 is a respected standard even though it is no longer mandatory. It is also worth noting that even though the software is currently being developed, by the time it is fitted to the UK Merlin much of it will be considered COTS.

For the current project, Lockheed Martin has prepared a MCSP Software Standard (CSS) which takes the elements of DO-178B, Def Stan 00-55 and Def-Stan 00-56 and provides a single development process. All subcontractors will be expected to comply with the CSS.

The certification process for safety-critical software, which is entirely within the Westlands Helicopters work share, is based on an assessment of the supplier ability to provide software of the required integrity. The assessment of the suppliers will consider their experience, any product history, the supplier's processes and track record for safety related software, the proposed software architecture and software partitioning, and the proposed information to be provided. This information is then mapped to what is needed for certification and the IPT will determine what additional information (if any) may be required. It is not expected that a "blanket approach", e.g. develop using formal methods or undertake SCA will be stipulated. In the IPT's opinion, it is preferable for a supplier to adopt practices with which they are familiar rather than expecting them to use practices they have never encountered before and may have difficulty applying. In addition, should further information be required then a third party with the appropriate experience and expertise may be contracted to undertake the extra activities.

This approach has currently been endorsed by the Independent Safety Auditor (ISA) – Frazer Nash, and QQ(BD).

2.7.2.3 Future Strategic Tanker Aircraft (FSTA)

The Future Strategic Tanker Aircraft (FSTA) will be based on a civil aircraft (Airbus A330-200), which currently has full civil certification, but will be adapted to provide air to air refuelling. Responsibility for delivery of the FSTA and an accompanying safety case rests with the AirTanker Consortium (EADS, Rolls Royce, Cobham, Thales and VT Group). The IPT is taking a "hands-off" approach, leaving responsibility for the preparation of the RTS entirely with the supplier, although the IPT and AirTanker will jointly operate the flight trials.

FSTA aircraft can be thought of as falling into three fleets as follows:

1. A Core Fleet used predominantly on the Military Register for military operations.
2. A Shoulder Fleet that will be available for military use or, when not required by the military, used to generate revenue from short term third party use. Shoulder fleet aircraft may operate on either the Military or Civil register, as appropriate to the operation being undertaken.
3. A Crisis, Tension and War (CTW) fleet that will remain on the Civil Register generating revenue whilst being operated by third parties on long term lease, but which will be available for military use at times of heightened demand.

The approach for certification is to rely on the existing certification activity as far as possible. The existing safety case will need to be assessed to determine sufficiency and it is expected that there will be a requirement to carry out some additional testing, but it is expected that this will not be extensive.

The rationale behind the strategy is that the operational profile of the aircraft is not significantly different to the profile which has already been certified. The intention being that AirTanker will obtain Supplemental Type Certification from EASA for the design changes required to the Airbus A330-200 to convert it into the FSTA.

The refuelling aspect is bespoke, and the AirTanker Consortium will be responsible for developing a Safety Case for this functionality. There is strong confidence in the ability of Airbus to produce sound software in accordance with the requirements of DO-178B.

A divergence from other IPT's is the appointment of a safety assessor other than QQ(BD). Instead, the Instituto Nacional De Technica Aeroespacial (INTA) from Spain is being consulted, although input from QQ(BD) is still being sought. This appointment has been accepted by the Release to Service (RTS) Authority.

2.7.2.4 A400M

The A400M is a new transporter airlifter to replace the Hercules. The project is being managed by a European Consortium OCCAR (Organisation Conjointe de Coopération en matière d'Armement) to interface with the contract supplier Airbus.

It is worth noting that the other bidders for the project (Lockheed Martin and Boeing) both proposed to develop the software in accordance with the requirements of Def Stans 00-55 and 00-56 while Airbus proposed its own procedures based on DO-178B.

The proposed strategy for gaining certification is to undertake 4 audits of each of the LRU suppliers, focusing on key areas which will be highlighted by a hazard analysis to be carried out by the IPT. The IPT will undertake Operational and Support Hazard Analysis (OSHA) and development of a hazard log database. It is believed this should be sufficient as Airbus is familiar with developing software to DO-178B and has a good track record in software development.

QQ(BD) have put together a target RTS, and populated it with the information that is currently available. This will be updated at regular intervals to ensure that the evidence is not diverging from the target RTS. Should significant divergence be seen, then consideration will be given to undertaking additional activities which may include testing or SCA. There is no current proposal to undertake such activities, and this has been accepted by QQ(BD).

The method of certification is for EASA to certify the baseline aircraft, with military deltas (such as low level flying) being certified by the Certification and Qualification Organisation (CQO). The CQO represent the 6 European nations who are interested in the A400M, and have established a Certification and Qualification Committee (CQC) to advise the A400M certification process. The CQC includes QQ(BD) experts.

2.7.2.5 Hercules C-130J

The Hercules C-130J programme is moving towards the Block 6 upgrade. Version 6.1 specifies the additional requirements to meet the UK, Danish and Australian markets. For the UK market, one such requirement is certification against UK military regulations. The Hercules IPT is trying to determine a sensible approach for certification, based on recommendations from QQ(BD), who are acting as the safety assessor for the programme.

One of the areas of concern is software. Although there is confidence in the software being developed by Lockheed Martin, some software development is being subcontracted. Defense Contract Management Agency (DCMA), on behalf of the US Government (USG), has undertaken audits of Lockheed Martin and is confident that SCA will not be necessary. It is understood that Lockheed Martin have used the experience gained from SCA undertaken in the 1990's to develop their software processes and procedures. However, it is not known if subcontractors have been audited yet. Visibility of the subcontractors' processes and outputs is limited, and, therefore, there is little awareness of the approaches being taken to ensure robust software is being developed.

STANAG 4107 "Mutual Acceptance of Government Quality Assurance and Usage of the Allied Quality Assurance Publications" has been called upon as part of the contract, which limits the UK MoD from undertaking additional auditing of the development programme. Therefore the IPT will only be able to rely on the audits undertaken by the DCMA.

The IPT have looked at the requirements of JSP 553 "Military Airworthiness Regulations", and have developed a compliance matrix of evidence available against the requirements of JSP 553. A gap analysis has revealed that the processes at Lockheed Martin are closely aligned with JSP 553. However, the problem remains of visibility of subcontractors. It was noted that there is no formal Safety Management Plan for the project, although the compliance matrix should form a significant input to this plan.

Although there was no contractual requirement on Lockheed Martin to prepare a Safety Case in the original contract, there is still a requirement to collate evidence of safety. The USG is reviewing the aircraft hazards and their mitigation, and outputs from this work are available to the IPT. In addition, the IPT will review the military deltas for the UK market and determine what additional evidence is required. The IPT is in a position to request that Lockheed Martin provide this data. The IPT believes that some SCA may still be required for subcontractors' code. The use of SCA is believed to be actively encouraged for this programme.

2.7.2.6 Sentry

The scope of the discussions was limited to the cockpit area. A particular upgrade of the display screens to flat panels was covered in the discussions. Although the contract for the flat panels included a requirement to meet DO-178B, it is not believed that this was flowed down to the subcontractors. Statements were made to the IPT that the software had been developed in the "spirit

of' DO-178B, but lack of visibility of subcontractors' processes and procedures meant that no assurance of this could be obtained.

In order to gain certification, an evaluation of the data available was undertaken. As access to the source code was denied, it was not possible to undertake SCA. Instead the IPT undertook a hazard assessment of the system, and determined the worst case scenarios. At the time of the procurement of the flat panels, Safety Cases were not required so one was not developed.

2.7.2.7 Chinook Mk 3

The Chinook Mk3 suffered many difficulties in its procurement resulting in clearance to fly the eight aircraft (seven aircraft delivered to the UK and the one remaining in the USA) not being granted. One of the key difficulties was the lack of an acceptable Safety Case for the helicopter. The current Chinook programme consists of removing and upgrading much of the avionics equipment on the helicopter.

The original Chinook Mk3 programme failed due to an absence of evidence of safety of the software. In addition, the IPT was unable to generate the evidence as access to the source code for the software of many of the LRUs was denied. There had been no requirement laid down in the contract to provide access to the source code (originally established in 1995) and therefore contractual problems prohibited completion of the Safety Case. In addition, where investigations of some software items were undertaken (e.g. the FADEC – Full Authority Digital Electronic Control) major issues were identified. The FADEC software had been developed in Assembly and was not amenable to SCA. The software, developed in the 1970's, was considered to be badly written, but of course standards and guidelines for software development at that time were not widely available. To address these problems a full replacement programme is now underway.

More confidence is now being placed in the current contract with Boeing to develop the necessary safety assurance required. The software will be developed in accordance with DO-178B. The IPT has developed a Safety Management Plan for the Chinook programme which provides guidance on assessing the integrity levels for complex programmable elements. It is envisaged that no complex programmable element will be classified with the highest integrity level. The Safety Management Plan requires that Boeing will prepare an Evidence Assurance Plan (EAP), to comply with the Software Certification Plan (SCP), for each complex programmable element. Each EAP will define how the evidence to gain software certification is to be collected. Any gaps in the available evidence will require additional activities but these are anticipated to be system level assurance activities rather than SCA activities. Auditing of suppliers to Boeing will be a key feature of gathering safety evidence.

QQ(BD) have provided independent advice to Boeing during the development of the SCP, so it is expected that the proposed approach to certification will be acceptable to QQ(BD) and that a RTS recommendation will be forthcoming.

This procurement programme is slightly different to the other IPTs discussed in this report, as the Chinook IPT has more contractual influence over the software development programmes as the software is being developed specifically for the programme and is therefore not COTS. The Chinook IPT may have more power to ensure that the appropriate evidence for the safety argument is being generated during the software development programme, as opposed to trying to establish a safety argument post development.

2.7.2.8 C-17 Globemaster

The Boeing C-17 was originally bought off the shelf from the US and put into service in the UK in 2001. Much of the software on this aircraft was developed in the early 1980's and 90's so did not follow the guidelines of DO-178B. Therefore, although a Safety Case for the aircraft was prepared, it had to provide mitigations against the very real possibility of software failures. Some of these mitigations have been in the form of operational limitations (e.g. no low flying).

Today, all new C-17s rolling off the production line have to comply with current best practice, and therefore, DO-178B is now applied to all new systems and changes made to the software on the aircraft. This has an impact on the UK programme, since any replacement LRUs will be developed to DO-178B.

The original safety case for the aircraft recognised the potential weaknesses with the software. A special authorisation from the Secretary of State was needed, which granted authorisation to deviate from the requirements of Def Stan 00-55, to enable the aircraft to fly. The current re-development of the LRUs, which includes developing the software in accordance with DO-178B, is a huge improvement to the aircraft and significantly strengthens the existing Safety Case.

The approach to gaining certification varies for each LRU, for example.

- The engines (FADEC) are certified to civil standards (as used in the Boeing 757 civil certification) and this is considered acceptable for the C-17 application.
- For the flight control a mechanical system is overlaid and so a lower integrity level was acceptable for the software, as it was considered that the mechanical flight controls provided adequate mitigation.

It is not possible for the IPT to undertake any post development analysis, as the IPT has to accept the development undertaken under the US contract. Any post development analysis is considered to be time consuming, costly and unlikely to provide any benefit.

One issue raised was that, historically, there has been a tendency to “tailor” standards to meet the suppliers/government needs. It was noted that although there may be a requirement to develop to DO-178B level A, an agreement may have been reached between Boeing and US government (based on a cost benefit analysis) to develop to a lower integrity level. More recently, QQ have been providing

advice to the project through the future software development programmes and have seen a significant improvement in the process adopted. This may have been driven by the need to satisfy GATM requirements, since the aircraft will be flown in civil air traffic lanes so it will have to demonstrate to the FAA that sufficient levels of safety have been achieved.

2.7.2.9 Future lynx

For the Future Lynx helicopter the only equipment on the aircraft which is required to achieve the highest level of safety integrity is the FADEC. The software within the FADEC has been developed and certified by the FAA to the highest DO-178B integrity level. However, given historical events, there was concern over the FADEC and initially SCA was advocated as the approach that would have to be followed to gain certification of the FADEC in the Future Lynx. Questions were immediately raised by the IPT and the Assistant Director/Airworthiness Design Requirements and Procedures (AD/ADRP) of whether there was a justification for carrying out SCA. The FADEC was a different system to that in the Chinook⁹ and it has many thousands of service hours, so already has some pedigree. AD/ADRP proposed an alternative approach of re-certifying the FADEC by a reputable third party, EASA. It is believed that this approach is acceptable to the Future Lynx independent assessors, QQ(BD).

There is still question over whether re-certification is necessary, since the operational profile of the Future Lynx is unchanged and the military deltas for the Future Lynx do not impact the FADEC in any way. If the evidence to support the FAA certification is available and sufficient then there should be no need to re-certify the FADEC in this manner. However, it is not known at this time if the evidence is available and sufficient. The cost of this re-certification is estimated as a few hundred thousand pounds. This amount represents a cost significantly less than retrospective SCA, but still not a trivial cost to the programme.

2.8 Initial IPT Feedback

A meeting was held on the March 7th 2006 at the DPA, Abbey Wood, where the findings reported in the first issue of this document report were outlined by ERA and some of the certification issues highlighted by the study were discussed. Most of the IPTs were represented including Sentry, Globemaster, FLynx, Merlin CSP with the MoD Sponsor and a guest from industry EADS.

Some of the initial views expressed by the IPTs were the following:

1. Use of DO-178B in general

The consensus was that the use of DO-178B in itself was not a particular problem, if used appropriately in the context of system development and assessment. In principle, the design

⁹ The FADEC has been implicated as a cause of the Mk2 Chinook accident in the Mull of Kintyre in 1994.

assurance evidence that DO-178B generated could be used for some of the software aspects of the safety case. The meeting accepted that there would be inevitable technical debate about the extent of the evidence required for the software safety case but, placed in the right context, compliance to DO-178B will be relevant and support the safety argument. (N.B. The meeting thought that the detailed criticisms of the DO-178B guidelines that have been made by safety experts and assessors were not “showstoppers” and would be addressed).

The rest of the meeting was about the “software aspects of certification” or acceptance of software rather than the details of the guidelines. DO-178B was used as an important example but could be replaced by any generally accepted software development guidelines.

2. Applicability of DO-178B certification (recertification)

The meeting agreed that compliance to DO-178B alone should never be enough evidence for acceptance of the same software in another application. Its system safety assessment may result in different safety requirements and the operational profile may require different testing. For example, there may be new hazards or hazard details for the application which may impose new safety requirements and /or different safety integrity requirements on the software. Furthermore, the whole software must be analysed to reveal any possible contribution to the new hazards.

3. Accessibility of relevant software and safety data

In the context of re-certification addressed above, access to software design artefacts may be required by technical experts. For example, a safety analysis of software design and source code may be required to show that failures cannot result in hazards associated with a new application. It was noted that most of these artefacts originated in the US and were increasingly the outputs of processes using DO-178B. All the IPTs expressed concern about obtaining access to these artefacts on reasonable terms in realistic timescales.

There was a long debate about accessibility. One suggestion was that a US Designated Engineering Representative (DER) could be appointed to make a technical assessment and declaration without revealing any technical details. However, it was pointed out that the DER would have to understand the system context, the UK system safety work and assessment approach. The suggestion reflected the frustration expressed in the meeting that a difficult problem required a new approach.

4. Civilian System with Military Deltas

There are an increasing number of systems originally developed for civilian use that have been slightly modified for military use to reflect the environment and operational requirements. The meeting agreed that one method of accepting these systems was to

concentrate safety assessment effort on the modifications or “military deltas”. Two important references were identified as:

- The Military Certification Seminar of September 2005 sponsored by ADRP, and
- The FAA Advisory Circular of December 2004 addressing Software Re-use.

5. Guidance required by Industry/IPTs for developing and accepting software

For developers, it was pointed out that guidance for the development of software was under review to support Def Stan 00-56 issue 3.

As regards the IPTs, there has been little software guidance to support the safety policy developed by the DASC and ADRP. The policy has concentrated on the requirement to introduce formal Safety Management Systems to develop and maintain Safety Cases based on hazard risk management with the RTS documentation linked to the hazard management.

Guidance is required to ensure that the software aspects of the acceptance of systems are linked to the Safety Management Systems that have been introduced. Three problems were raised:

1. Certification consistency for software

It was perceived by industry that IPTs did not have a consistent approach (although programmes are inevitably at different stages of the lifecycle). One problem was the rate of adoption of new standards and imposition on industry. The status of Def Stan 00-55 as “obsolescent” is still causing confusion for ongoing programmes.

2. Reconciliation of software safety approaches

There is a perception from IPTs that there is no consistent agreement about software safety, particularly between countries and industrial sectors. For example, for a SIL 4 military system in the UK, an ISA may reasonably require (previously backed by Def Stan 00-55) that formal methods be used to specify the software requirements and to formally demonstrate their refinement through design into code.

If the software was produced for a US military application or in the civil aviation sector, then formal methods may not have been used, although the highest criticality or design assurance requirements may have been assigned. The US developer may claim (correctly or incorrectly) that more effort has been directed into analysing the ways in which the software can fail dangerously and analysing its interaction with other systems and the operators.

There is still an ongoing debate between this “system safety” approach exemplified by the US military and the UK approach which claims that formal methods help to clarify the software requirements and ensure that they are implemented correctly.

(N.B. This debate has been raging for at least one decade if not two. It may not be possible for the approaches to be reconciled or one method to be substituted by another. Outside military applications, it has been known for an assessment organisation to change its safety engineer for another who demands a different safety approach from the prime contractor which results in rework, consequent overspend and court cases).

3. Efficiency of RTS production and traceability of RTS contents

It was surprising that this was raised as most IPTs produce RTS documentation efficiently and link limitations back to the Hazard Log. The comment reflects that guidance or best practice should be captured for new IPTs.

3 Use of Static Code Analysis

3.1 Introduction

Static Code Analysis (SCA) is any analysis of code (software) performed without actually executing that code. It is complementary to dynamic code analysis, which does involve execution, e.g. testing. One of the key benefits of undertaking static analysis is that there is no need to simulate the operational environment as the software is not executed. Static analysis is used for a vast range of reasons and techniques range from:

- Manual or partially automated techniques; such as code walkthroughs and inspections
- Fully automatic techniques, which are typically limited in their power
- Comprehensive techniques, in principle very powerful, but not fully automated.

SCA can be undertaken at various stages of a software development depending on what procedures and tools are adopted. SCA can be conducted at the time of compilation of source code using facilities available in the compiler tools, for example, PC-Lint finds syntax errors in C programs based on recognised standards for the development of C code. SCA can also be carried out during testing. A dynamic testing tool which is widely adopted in the military development environment and includes some capability for static testing is LDRA TestBed. This tool is available for a number of programming languages and includes a range of static analysis capabilities. Other static analysis tools available include MALPAS and SPARK Ada. These tools go further than LDRA TestBed enabling proof of correct implementation of requirements.

For the purposes of this report, the SCA of most interest is retrospective (i.e. post-software development) SCA to assess the safety of software (and, possibly, subsequently achieve greater safety assurance).

Retrospective SCA for safety typically includes some or all of the following activities:

- Manual code inspection and comparison to requirements
- Highly automated syntactic analyses such as data flow and information flow
- Semi-automated semantic analyses
- Semi-automated correctness analyses.

3.2 Past Experience

In the UK military environment, SCA is often thought of as the post development, program proof activities, such as that carried out for the Hercules C-130J development. This sort of analysis will be referred to as retrospective SCA.

The rest of this section refers to two papers on the subject based on the results of one programme of work, the retrospective SCA carried out for the Hercules C-130J in the 1990's. Therefore, these papers are based on a (very) small, non-random sample and hence the conclusions cannot be generalised.

The first paper was written in 1999 (Ref. 18) to disseminate the activities undertaken on the SCA of the C-130J. The paper notes that one of the major difficulties encountered was the "acquisition of data from the suppliers", one of the key problems identified across the range of IPTs interviewed for this study.

In reviewing the errors identified by the analysis, it is noted that only about 6% of the 11,590 entries in the anomaly database were classified as errors "which could threaten safety" either on their own or in the presence of other errors. However, in reviewing the results of this activity, no assessment is given on the maturity of the source code before subjecting it to SCA, the experience of the software suppliers in developing safety critical code in accordance with the requirements of DO-178B, or other key factors which may impact on the desire to carry out SCA, and the benefit that may be gained. It is likely that current adoptions of better working practices for software development mean that the number of software errors identified as a result of retrospective SCA is significantly reduced.

In addition, the paper noted that although a range of analysis techniques were adopted (including information flow, control flow and semantic analysis) most of the errors (60%) were found during familiarisation. Familiarisation did not involve any analysis using toolsets but was undertaken at the start of each analysis activity to gain understanding of the code and supporting documentation by manually reviewing the supplied information and undertaking a code walkthrough activity. The activities which constituted part of the "familiarisation activity" can be undertaken by competent individuals at any stage of the development to assess the quality of the software. Indeed, current best practice does tend to require many of these activities, such as peer reviews, code walkthroughs or independent assessment, be undertaken as part of a typical good quality development process. Therefore, although SCA may be advocated, it may be that a scoping exercise be undertaken initially through "familiarisation" activities such as review, audit and code walkthrough to gain a level of confidence in the software being developed. This view is supported by the QQ paper "Software Static Code Analysis Lessons Learned" (Ref. 19), written more recently in 2003, which notes that independent code walkthroughs "are the most effective [SCA] technique for software anomaly removal". Should the IPT undertake such a review activity, and from it identify few errors and strong evidence of adherence to standards is seen, there may be an argument for not carrying out further more complex, retrospective SCA.

The QQ paper has determined some costs¹⁰ in undertaking retrospective SCA, and quotes a typical cost of \$13 million per aircraft type. The paper considers this to be cost effective since the cost

¹⁰ The source of these costs is not discussed.

represents approximately 0.4% of typical development costs. However, many current UK MoD procurement programmes are procuring off-the-shelf programmes with much smaller budgets (A400M is quoted as costing £18 million), therefore a cost of \$13 million (approximately £7.5 million) is prohibitive, and representing a much bigger percentage of the overall budget (approximately 9% of the budget for A400M). However, it is worth noting that an argument not to do something based purely on cost is not, on its own, justifiable. To fully justify not carrying out retrospective SCA, application of the ALARP (as low as reasonably practicable) principle is required to determine whether there is an argument based on cost and benefit for not carrying out the activity. However, retrospective SCA is clearly costly, and it is likely that alternative approaches are likely to be more cost effective.

The QQ paper discusses the need for some analysis activities other than relying solely on testing, as errors have been found in software that has been “rigorously” tested. The paper claims that “surprising amounts of dead code have been found in code developed to RTCA DO-178B Level A and B”. However, the emphasis remains on retrospective SCA as a solution. Although, the conclusions do suggest that SCA conducted in line with the development is an effective software analysis technique and recommended in the context of safety-critical software, little further is discussed. Such applications of SCA techniques provide an immediate benefit, identifying software coding errors early in the software development lifecycle, enabling corrections to be implemented in a timely manner which in turn reduces development costs.

The QQ paper also assesses the impact of the programming language used in developing code on the number of errors found, through assessment of metrics on the number of errors found during “various” programs¹¹. C is considered poor with a worst case of up to 500 anomalies identified per 1000 lines of code (loc). SPARK is significantly better with approximately only 4 anomalies per 1000 loc.

In conclusion, there are many factors which could impact on the number of errors that may be encountered in tested code. Some of this relate to the processes used for the development, the experience and competence of the engineers undertaking the development activities and the choice of programming language. In the past, there has been little confidence in some of these aspects, and evidence has confirmed that many errors have remained in tested code. However, much progress has been made in the availability of guidance material and the competence of software engineers such that the quality and robustness of source code is continually improving. This could mean that the ability to undertake retrospective SCA is much improved, since some of the past difficulties such as amenability of the source code to SCA may be overcome. However, the benefit of undertaking retrospective SCA under these circumstances is put even more into question. Retrospective SCA is evidently expensive and time consuming, and may not be justifiable as an approach for the typical programmes being undertaken by the current IPTs.

¹¹ The source of the figures is not provided in detail.

4 Conclusions

4.1 Variation in the Quality of Application of DO-178B for Certification

The study was made aware that there has been variation in the rigour of the application of the DO-178B guidance on different projects, important quality factors being identified as differences in the competence of the software engineers and their understanding of safety. Furthermore, the interpretation of DO-178B for acceptance by the Designated Engineering Representatives (on behalf of the regulatory authorities) is also considered to have been variable. It is not clear whether this latter variation is due to differences between DERs or the evolution of interpretation by the regulator.

Nevertheless, a more consistent application and acceptance regime for the guidelines has been developing in line with increasing understanding of software engineering, its relationship with avionic safety, and the maturity of the guidelines.

4.2 Interpretation and Extension of DO-178B

The re-certification of software for a different application should start by examining the actual safety programme under which the software was originally developed and certified. For several companies, the guidance of DO-178B has been interpreted and enhanced with other standards and guidelines to define a complete process for safe software and system development. These internal procedures integrate the software development with the system safety management process, which is (intentionally) omitted from DO-178B.

4.3 Some issues for High Integrity Software

Errors in the software requirements specification are well known to be a significant, if not the major, cause of accidents in safety-related systems; that is, more accidents were found to be caused by getting the requirements wrong than by incorrectly implementing them.

From the US industry survey, it appears that requirements definition in accordance with DO-178B may not be sufficiently robust to prevent errors being introduced. DO-178B describes system requirements being allocated to software by the system development process along with an associated Software Level; there is an implication that the requirements are defined by system engineering and “handed over” to software engineering for implementation. DO-178B does require the software requirements to be reviewed and, where appropriate, analysed for accuracy, consistency and completeness. Within the software and safety engineering communities, there is a recognition that the process of the derivation of software requirements needs more guidance as significant safety-related errors in the software requirements have been found to be the contributory source of several accidents and incidents.

The intention behind Def-Stan 00-55 was to provide methods and notations for software requirements’ definition to eliminate ambiguities and inconsistencies by encouraging clarity of

expression. The use of formal notations forces the specifier to be explicit and clear and enables simple syntax and type checking of requirement. Furthermore, formal methods can be used to formulate safety properties that the specification should embody which can then be mathematically proved. Such methods are not strongly emphasised in DO-178B but they can be introduced as special analysis required for the software requirements data in the Software Verification Plan.

This aspect highlights a difference between software safety cultures in the US and Europe. In the US there is a tendency to develop the safety software with less formality and test rigorously, whereas in Europe there is more emphasis on analysis and review of the requirements or design to ensure correctness before moving to subsequent phases of development. This is reflected in the stronger focus within DO-178B on testing. Although DO-178B includes objectives for review and analysis there is little guidance to support the developer in the conduct of these activities. However, there is more guidance provided on testing. Outside the US, there is a stronger reliance on a diversity of evidence from testing and other techniques such as analysis or formal proof to provide an argument for safety.

4.4 Quality and Perception of DO-178B Software

There has been concern about the quality of software written to DO-178B arising from some projects at low integrity levels for DO-178, DO-178A and the early days of DO-178B. Although most of this evidence is “hearsay” and the same examples are repeatedly quoted; there were undoubtedly some poor quality software development for which the system achieved certification. Software engineering in general is now much better understood as a discipline and the general quality has increased.

Nevertheless, there is little hard evidence collected or agreement about the quality of DO-178B software, in particular frequency of software failures that contribute to accidents or incidents. It is claimed that a software failure has never been the primary cause of the loss of an aircraft but there are many accidents and incidents in which software failure has been contributory.

There is a disparity between the IPTs views on civil certification based on the use of DO-178B is adequate for current procurements. Some IPTs are prepared to accept the existing certification with minimal review, while others will take very little account of this and request the contractor to collect the evidence and present an argument from scratch. There is some convergence in the IPTs approaches such as concentrating effort on military deltas, but there may be scope for reducing the effort further if there was a greater confidence in software developed to DO-178B.

The IPTs are fairly unanimous in their desire to avoid the use of the more rigorous retrospective SCA methods, based on a belief that they are costly, time consuming and provide little added benefit. Much of this view arises from the reports of the experiences gained in conducting a retrospective SCA as carried out for the C-130J. However, there are a whole range of safety analysis techniques that can be carried on the design or code that are significantly less costly and likely to bring much added benefit. Indeed, many widely used compilers and testing tools now include some aspects of static

testing by default. However, it is worth noting that an argument not to do something based purely on cost is not justifiable.

4.5 Content of Certification of Software Systems

Finally, it is noted that the role of how software has been certified in a US civil application does not seem to be widely understood, and there is much benefit in ensuring that there is a wider appreciation on the context of software being certified as part of a system and not standalone. The IPTs need guidance on how to take the existing certification evidence and the approaches that could be followed in order to gain certification of their product in a UK military application. Currently, each IPT is developing its own approach, customised for each project. Some commonality amongst the approaches would be beneficial.

Therefore, the following section proposes an initial way forward. Reference is given to existing research programmes which are establishing better ways to develop safety arguments for UK military applications. However, in order to develop these ideas fully to provide useful guidance for the IPTs, further effort is required.

5 Proposed Approaches to Certification

5.1 Introduction

The purpose of this section is to outline approaches that could be adopted for addressing the software aspects of certification. An important conclusion of the study is that the software cannot be addressed in isolation of the overall certification process to which it contributes. In moving from the use of DO-178B software programme in a civil or US military system to its contribution to the System Safety Case for UK military certification, it is important that the previous system safety assessment is fully understood. This section is split into three to address the software aspects of certification of the following:

1. new programmes requiring bespoke software (N.B. this software may be layered on top of COTS components such as Operating Systems which have been incorporated in several, or in some cases many, different certified systems)
2. programmes utilising software already developed to DO-178B
3. programmes utilising software developed to other standards.

These approaches are only outlined. Further work is required to fully develop one or more of these approaches and undertake a trial on a real programme to ensure the applicability of the approaches to real projects.

5.2 Common Standards

The policy for defence aviation safety is provided by the Defence Aviation Safety Board (DASB) which sponsors the JSP 550 series of documents which establish military aviation policy, regulations and directives across the services. In particular, the Military Airworthiness Regulations (JSP 553) set out the requirements for the safety management of an aircraft platform including the “Release to Service” process by which a new aircraft or a modified aircraft is handed over to the service with operating limitations and maintenance procedures for safe operation. JSP 553 is maintained by the ADRP (Airworthiness Design Requirements and Procedures) department of the DPA.

Def Stan 00-56, “Safety Management Requirements for Defence Systems” describes the requirements for safety management of a specific project, such as the development and introduction of a new system to the air platform providing additional capability. The standard describes requirements for the IPT’s and Contractor’s Safety Management Systems and the integration of subcontractors into the safety programme.

The JSP 553 standard provides the aircraft domain context for the application of Def Stan 00-56. The scope of Def Stan 00-56 includes the requirements for the management of safety aspects of the design at system and subsystem level and including hardware and software.

5.3 New Programmes

Historically, an IPT has directly influenced the system level safety management system through the prime contractor to ensure that the safety argument is made at the system level. The contractual chain has been used to obtain access to Supplier held evidence to support the safety argument. In general, an IPT has not directly influenced how the Supplier developed software but the requirement to comply with Def Stan 00-55 has been imposed by the prime contractor on behalf of the MoD.

For new programmes, the IPTs have stressed the limitation of their potential influence on the software development processes by the following factors:

1. the fact that many of the LRUs are existing or evolutions from existing products
2. the reluctance of suppliers to develop bespoke products for the small UK defence market using processes which they perceive to be “non-standard” and involve greater risk.

However, now that the cost of software development dominates the cost of system development, the software development processes of the Suppliers have the potential to influence the system level safety management system or at least pose the question about how they can be accommodated. As regards processes compliant with DO-178B, this potential is given support by the perceived success of civil aviation safety.

5.3.1 Mapping Civil to Military Standards

This section describes how the civil aviation safety standards (and their associated safety development and assessment requirement) can be mapped to the military aviation standards (with their differing requirements for safety development and assessment). It should be emphasised that the operational context for military use may be similar or significantly different from civilian use.

Table 1: Civil and related Military Standards below identifies the important documents relating to the Civil Certification of software systems in column 1.

The document(s) containing similar information that could be used for military certification are shown in column 3 of the corresponding row. For example, the first row shows that the civil regulation FAR/CS 25.1309 is replaced by sections of JSP 553 and Def Stan 00-56 Part 1.

Subpart 1309 is used as an example as it addresses the general requirements for “Equipment, Systems and Installations”. For specific equipment and systems such as autopilots, these requirements are complemented or can be overridden by the other subparts which deal with these systems in detail.

Civil Development	Comment	Substituted by	Comment
FAR/CS 25.1309	Equipment, Systems & Installations	JSP 553 DS 00-56 Part 1	Overall Safety Target
AMC 25.1309	System Design and Analysis	DS 00-56 Part 1 DS 00-56 Part 2	System Assessment Process
ARP 4754	Complex Avionic Systems		No equivalent
ARP 4761	System Assessment Methods	ARP 4761	System Assessment Methods-Replacing 00-58
DO-254	Complex Hardware	DO-254	Replacing 00-54
DO-178B	Software	DO-178B	Replacing 00-55

Table 1: Civil and related Military Standards

In the table, the guidance of DO-178B for software and DO-254 for complex hardware are suggested as being used for the military development. The use of DO-254 for the civil aircraft has recently been supported by an Advisory Circular and already its importance has increased in the US. It addresses Programmable Logic Devices (PLDs) such as FPGAs and Structured ASICs and is closely related to DO-178B.

As regards rotary aircraft, the relevant chapters of the regulations are FAR/CS 27 and 29 for normal and transport category “rotorcraft” respectively. The numbering systems for subchapters are consistent with FAR/CS 25 so both 27.1309 and 29.1309 deal with “Equipment, Systems and Installations”. As regards the acceptable means of compliance for “rotorcraft”, the FAA have rationalised their guidance into two large documents each over 1000 pages which include a detailed description of the required safety process and safety analyses and place DO-178B in context. These are Advisory Circular 27-1B and 29-2C issued in December 2003 and referenced directly from the corresponding European AMCs 27.1309 and 29.1309.

It is noticed that there is no specific guidance for system development other than Def Stan 00-56, which is targeted at the safety management system. This is a potential weakness that should be addressed. There is civil guidance in ARP 4754 for complex highly integrated systems and the ASAAC draft standards do address important system architectural aspects such as IMA. Most US military systems are developed using Mil-Std 882.

These system standards should be used in conjunction with appropriate standards for component development, such as DO-178B for the software and DO- 254 for hardware.

Any new development needs to take account of potential weaknesses in the development process covered by DO-178B. One such potential weakness that has been identified is its lack of formality

with respect to specification of requirements. This could lead to the introduction of errors during the requirements definition phase of the software lifecycle. To address this possibility, QQ and the University of York are undertaking a programme of work (Ref. 28) to establish how a more formal approach to the software development can be undertaken, which satisfies the objectives of DO-178B, but also provides the safety evidence needed to develop a Safety Case in accordance with the requirements of Def Stan 00-56 (Ref. 2).

The QQ and University of York approach looks sensible and pragmatic with one important proviso: the tool sets and notations to support the approach must be useable and understandable by other stakeholders and particularly system engineers as well as the software engineers. The problem of the validation of software requirements lies at the interface of system and software engineering and a common understanding is essential; clarity for the software engineer must not be at the expense of the ability of other stakeholders to check the specification. With this proviso, the approach should prove to be applicable to the MoD and provide them with a solution for new development programmes.

5.4 Existing Developments to DO-178B

The certification of software that exists and has previously been developed to DO-178B for use in a military application remains an area where further guidance is required. There are two possible scenarios which need considering:

1. software that has been developed to DO-178B and certified for a civil application
2. software that has been developed to DO-178B but not certified for a civil application.

5.4.1 Certified software

As stated in Section 2.5 of this report, software is never certified as a stand alone item, but always in the context of a system application. In order to gain certification, a DER for the FAA is appointed to determine whether the system has been developed in accordance with the appropriate airworthiness regulations, in particular FAR 25.1309 (Ref. 4). These regulations require that an assessment of the safety of the system be undertaken to demonstrate that the system and its components are sufficiently robust.

The DER is responsible for approving, or recommending approval of, technical data to the FAA. The DER will not only undertake audits and review technical information of the software development process but also consider the system safety assessment process that has been adopted to address the risks associated with the system. These reviews are undertaken throughout the development programme and of the final safety justification before the DER will provide the FAA with a recommendation.

Therefore, any system that has been certified by the FAA, will have developed a system safety justification. In addition, the regulator through the DER has established the independent acceptability

of this justification including, where applicable, the software development process and its application. For the purpose of re-certification for a UK military application, the IPT needs to establish how much of the existing safety justification can be re-used. The extent of re-use depends greatly on the differences between the applications. If the system context is largely unchanged then it is likely that the hazards of the system are similar and that the contribution of software behaviours and failures to the system hazards are also likely to be similar. For any differences in system operation a reassessment of the risk is required. It may be that there is an increase in the risk, but this increased risk may be acceptable for that different application.

In broad outline, it is suggested that the approach to re-certification of software in a new application should be as follows:

1. Establish the differences between the current application and the existing application (these differences are often referred to as military deltas). For many of the IPTs that have been interviewed for this study, their use of the software components is not significantly different, although there may be major difference in the operational profiles of the aircraft.
2. Undertake a safety assessment to determine how anomalous software behaviour may impact the hazards associated with the new application.
3. Establish the risks associated with the use of the software in its current configuration. Is this risk acceptable? It is possible that an increase in risk is acceptable for the military application.
4. If the risk is acceptable then no further modifications are required.
5. If the risk is unacceptable then, as in any other safety management system, a risk reduction exercise shall be carried out. As a result of the risk reduction exercise, there may be a need to make modifications and changes to the software (if this is possible) or the system, or there may be a need to carry out additional analysis and/or testing to provide a better understanding of the risks associated with the system.

Finally, the IPT will need to develop an application specific Safety Case, and will need access and understanding of the context in which the existing application has been certified. It is not appropriate to accept the FAA approval process without this understanding, although significant confidence is provided by the knowledge that an approval has already been granted. Indeed, the FAA recognises that it is not appropriate to reuse software without revisiting the development, and has prepared advisory circular AC 20-148 (Ref. 20) “Re-usable Software Components”, which includes expectations from the certification authorities on the subsequent use of an accepted reusable software component. The advisory circular provides guidance on the software aspects of re-certification of a system re-using a software component from a previously certified system.

In order to populate an application specific Safety Case, the IPT will need to reference, as a minimum, the software lifecycle data. Adherence to the objectives and processes defined in DO-178B will generate a large volume of data which can be used to support a Safety Case argument. The application of DO-178B, along with the use of appropriate development tools, by experienced and competent software engineers is likely to generate software of sufficient quality to support an argument for the integrity of the software. There is no “guarantee” that following the process will generate correct software, but certainly adherence to a process is essential before any claim on the quality and robustness of the executable software can be made.

In addition, assuming that competent organisations are being used for the software development, the software developers are typically well aware of the avionic regulations to be adhered to, and it is recognised that these organisations often undertake work additional to the processes defined in DO-178B. By undertaking the additional activities a more holistic approach to the software development, taking account of risk and safety is more likely to occur. As part of the development of the proposed approach, it is suggested that competent organisations, such as Boeing and Airbus, are contacted to discuss the approaches they adopt to ensure an adequate development process.

This process for certification needs much more detail to provide sufficient guidance to IPTs for future development programmes. Further effort is required to support the development of appropriate guidance material.

5.4.2 Non-certified software

DO-178B is recognised as an appropriate guidance document for the development of software which is required to be of a high integrity. The guidance provides objectives that need to be met to ensure that a robust process is followed for the development, and provides some confidence that the software will perform as required. As such, there may be generic developments of software which are not, at the time of development, intended for a specific application, but the development has made use of recognised good practice as outlined in DO-178B. Future use of the generic product may be appropriate in a military application and an understanding is required for how certification can be achieved of the generic software product in such a military application.

Since the product will not have previously been certified there will be no formal certification evidence available to draw upon, although some evidence will naturally exist. As stated above, software engineering practices are continuously maturing and developers are more aware of their legal obligations, and therefore are likely to take account of potential risks during the software development. For a generic product, it is impossible to fully address safety as the hazards to which software behaviour and failures could contribute are unknown. Therefore, objectives of DO-178B such as ensuring traceability to system requirements and integration with the system safety process will not have been achieved. However, an understanding of how the software behaves under certain failure circumstances may have been established.

In this scenario, a full safety management process will need to be established to identify all of the hazards of the system and the potential contribution from the software to these hazards. Assuming that the software is being purchased “as-is” and that modifications to the software cannot be implemented, any mitigations to hazardous software behaviour will require design modification at a different level, e.g. at the system level. If the risk is unacceptable, in a worst case scenario, there may be a need to re-design and/or re-develop the software.

In addition, an evaluation of where the outputs from the DO-178B development process may be considered to be insufficient to support a safety case argument will need to be established. Additional analysis and/or testing activities may have to be undertaken. Some areas of weakness have been reported within this document, in particular including lack of formality of requirements capture and over reliance on testing. As covered above, it is possible that additional activities are already being adopted by the software developers but this will depend on the maturity and experience of the software developers. A software developer applying DO-178B for the first time, who sticks rigidly to the letter of the guidance material, may not develop software that is sufficiently robust for an avionics application. However, companies who have applied DO-178B for many years are likely to have adapted additional activities to ensure a more complete development.

As already suggested, an understanding of these typical activities is needed, to establish what current best practice is. Once this is understood, the benefit and cost of undertaking such activities post typical DO-178B development needs to be determined. This information can be used to propose a programme of work which can be applied to an existing development to ensure that the sufficient evidence of safety is available to support a safety case argument.

Further effort is also required to support the development of the approach to certification for software developed to DO-178B, but not certified.

5.5 Existing Development to Other Standards

For the purpose of this study, the certification of software developed to a standard other than DO-178B is considered to be outside the scope of the study. DO-178B is a significant influence on the development of avionic software for civil applications, and therefore remains a key focus for the study. Should other standards or guidance document be considered appropriate, then a review and impact of the processes described therein could be undertaken as a separate programme of work.

6 References

1. RTCA (and EUROCAE ED-12)
Software Consideration in Airborne Systems and Equipment Certification
DO-178B
December 1992
2. Def Stan 00-56
Safety Management Requirements for Defence Systems (2 Parts)
Issue 3
December 2004
3. FAA
Advisory Circular #20-115B
US Department of Transportation
January 1993
4. Federal Aviation Regulations 25.1309
Airworthiness Standards, transport category airplanes
Equipment Systems and Installations
1977
5. FAA
Advisory Circular: System Design and Analysis
25.1309-1A
June 1988
6. FAA
Reusable Software Components
AC 20-148
December 7, 2004
7. YSE
Comparison of Standards for Safety Related Software Development
CF171/3/53
ASSC/330/3/425
June 1999
8. Def Stan 00-55 (withdrawn)
Requirements for Safety Related Software in Defence Equipment (2 Parts)
Issue 2, August 1997

9. IEC 61508
Functional Safety of electrical/electronic/programmable electronic safety-related systems
2001
10. ARP 4754
Certification Considerations for Highly Integrated or Complex Aircraft Systems
April 1997
11. ARP 4761
Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne
Systems and Equipment
December 1996
12. R. A. Weaver (York University)
The Safety of Software – Constructing and Assuring Arguments
PhD Dissertation
September 2003
13. Def Stan 00-56
Safety Management Requirements for Defence Systems (2 Parts)
Issue 2, December 1996
14. NASA
Streamlining Software Aspects of Certification: Technical Team Report of the First
Industry Workshop
NASA/TM-1998-207648
April 1998
15. NASA
Streamlining Software Aspects of Certification: Report on the SSAC Survey
NASA/TM-1999-209519
August 1999
16. HighRely
Military Avionics Certification via DO-178B and DO-254
17. HighRely
DO-178B Costs Versus Benefits
18. K J Harrison
Static Code Analysis of the C130-J Hercules Safety-Critical Software
Aerosystems International
1999

19. A. German
Software Static Code Analysis Lessons Learned
QinetiQ
November 2003
20. FAA
Advisory Circular: Reusable Software Components
AC 20-148
December 2004
21. Boeing, Leslie Johnson
DO-178B, “Software Considerations in Airborne Systems and Equipment Certification”
1998
22. AMSC
Military Standard Software Development and Documentation
MIL STD 498
December 1994
23. Industry Implementation of International Standard
Information technology—software life cycle processes
ISO/IEC 12207
1995
24. DERA Malvern (C Newton and C Pygott)
A Comparison of Avionics Standards
DERA/CIS/TR990319
ASSC/330/2/167
25. Adelard
Methods for assessing the safety integrity of safety related software of uncertain pedigree
(SOUP)
337/2001
2001
26. Adelard
Justifying the use of software of uncertain pedigree (SOUP) on safety-related applications
336/2001
2001

27. ERA
Comparison of Defence Standard 00-56 and ARP4761/4754
ASSC/330/6/2
Issue 1, July 2002

28. A. Galloway, R.F. Paige, N.J. Tudor, R.A. Weaver, I. Toyn, and J.A. McDermid
Proof versus Testing in the Context of Safety Standards
in Proc. Digital Avionics Systems Conference (DASC) 2005, IEEE Press, Washington,
USA, October 2005.

Appendix A

Previous Research and Publications

A.1 SSAC (1998) – US Industry concern over DO-178B

Some concern over the burden of certification of systems with software developed to DO-178B in the USA was raised in the late 1990's. To try to address some of these concerns the FAA began the Streamlining Safety Aspects of Certification (SSAC) programme in 1998. The focus of the programme was to try to address concerns about the time and expense required for high integrity software. Three industry workshops and one FAA workshop were held in 1998 with wide representation. Unfortunately, funding for the programme was withdrawn and further work to determine the validity of the preliminary findings and to establish a plan for improvement was not undertaken.

The results of the workshops are published in the following reports:

- Streamlining Software Aspects of Certification: Technical Team Report of the First Industry Workshop (Ref. 14)
- Streamlining Software Aspects of Certification: Report on the SSAC Survey (Ref. 15).

The report arising from the first workshop notes that software engineering (at that time in 1998) is not considered to be a mature discipline and many questions remain about the relative effectiveness and expense of various software engineering methods and processes embodied in DO-178B. For evidence, examples are provided of software errors that have led to incidents and the report states that "The types of errors lead to concerns about the effectiveness of the certification procedures for this software, as embodied in DO-178B".

The first workshop sought the views of over 120 representatives to determine:

1. Are the techniques prescribed in DO-178B effective?
2. Can software aspects of certification be streamlined without affecting safety?
3. Are costs incurred that do not contribute to safety?

The comments raised at the workshop with grouped into two categories. Two further groupings within each category were used to classify the comments.

1. Issues that are not specific to DO-178B
 - Issues within the FAA
 - Issues within industry
2. Issues specific to DO-178B
 - Issues about the adequacy of guidance in DO-178B
 - Issues about the benefits of DO-178B.

Many areas of inadequacy in the guidance provided by DO-178B were highlighted. These included inadequate and ambiguous guidance for:

- documentation
- planning and configuration management
- requirements definition and analysis
- partitioning
- verification activities
- tool qualification
- COTS software
- reuse of certification data
- reuse of legacy systems
- non-airborne systems.

Under the category with respect to the benefits of DO-178B, comments were raised covering:

1. the extent to which DO-178B provides benefit beyond that provided by other industry accepted practices is unclear
2. the effectiveness of some specific activities required by DO-178B is unclear
3. DO-178B inadequately provides for innovation
4. DO-178B inadequately addresses the effect of software on the safety of the overall system.

The comments raised at the first workshop were used to undertake an industry survey based around these specific areas of concern. The results of this survey are documented in the second SSAC (Ref. 15) document identified above.

One of the areas considered was whether DO-178B adequately addresses safety. When questioned on this matter over 59% felt that it did. However, 49% also stated that they often conducted additional safety activities over and above those required by DO-178B, and 57% said they provided safety level information in addition to the software level.

The respondents were asked if they had “Ever worked on a system developed in compliance with DO-178B in which a software-related system error resulted in a service bulletin or Airworthiness Directive”. 28% of those questioned responded “yes”, and identified the source of these errors mainly from requirements (47%), design (32%) and coding (35%). Those who had not experienced safety errors were questioned on the benefit of DO-178B in contributing to the absence of such errors. Of those questioned, only 37% responded “yes” whereas 63% stated that DO-178B made little or no contribution to the absence of such errors.

A final objective in considering the safety aspects of DO-178B was to determine how industry treats derived requirements. DO-178B requires that derived requirements be subject to a system safety assessment (see Section item j Section 5.1.2 of DO-178B, for example). Only 9% claimed that this was done. Other techniques for dealing with derived requirements were undertaken.

In undertaking the survey, the SSAC raised the following recommendations to address safety:

Recommendation K1: The FAA should study the relationship between DO-178B and safety and the activities actually performed by the industry to ensure system safety.

Recommendation K2: The FAA should clarify compliance requirements and intent for derived requirements.

Unfortunately, funding for the SSAC was subsequently withdrawn and no further work undertaken by the SSAC programme.

A.2 Boeing (1998) – Comparison of DO-178B with MIL STD 498

Leslie Johnson of Boeing provided a “practitioner’s discussion” (Ref. 21) on DO-178B in 1998. The paper explored the difference between DO-178B and MIL STD 498¹² (Ref. 22) and highlighted some problem areas for those from a military culture. MIL STD 498 is a software development standard, and is not specifically aimed at the development of high integrity software, but provide a generic process for development of software of good quality.

The paper notes that “the objectives of these standards seem similar enough to be able to transition to commercial certification requirements without much difficulty. Yet, while there is similarity, there are fundamental differences in DO-178B that need to be understood by people familiar with MIL-STD processes”. The first difference noted is that “data for a military program is used to confirm understanding and support maintenance” while in the commercial arena “the customer uses the data as compliance evidence to the certification requirements”. The significance of this comment is not fully expanded.

The paper goes on to express that DO-178B is more rigorous in its requirements for evidence and verification. This is not surprising given that DO-178B relates to the development of high integrity software. DO-178B includes more strenuous requirements for documentation trail and evidence of completeness through review and analysis. In addition, DO-178B is more rigorous in its requirements for verification and traceability. Worryingly, the paper states that, in the US military field, software engineers do not “meticulously keep proof of the processes” since this is the responsibility of Quality Assurance. This is a practice which is not acceptable for the development of high integrity software,

¹² The Standard was withdrawn in June 1995 only months after being published and replaced by International Standard [ISO/IEC 12207: 1995](#): Information technology—software life cycle processes (Ref. 23).

and certainly any software developed in such a manner would not be approved by any DER so would not be certified for use on an aircraft.

A.3 YSE (1999) - Comparison of DO-178B with 00-55

A.3.1 Scope

In 1999 York Software Engineering (sponsored by the ASSC) undertook a comparison of DO-178B against Def Stan 00-55 (Issue 2). The findings of the study are published in Comparison of Standards for Safety Related Software Development (Ref. 7). In addition comparisons were also made against IEC 61508. Detailed comparisons on a clause by clause basis were made from one standard to another, enabling identification of requirements in DO-178B that are excluded from Def Stan 00-55, and vice versa. The following sections are taken from the report and provide the full details of the omissions identified in the documentation of interest to the current work programme.

References to sections of DO-178B are identified as RTCA *x* in the following sections, while reference to sections of Def Stan 00-55 are identified as MOD *x*.

A.3.2 Comparison of RTCA DO-178B against Def Stan 00-55

This section describes activities and data that are required by Def Stan 00-55, but are either not present in or not required by DO-178B.

Safety Management (MOD 5-11)

Safety Management activities are outside the scope of DO-178B.

Roles and Responsibilities (MOD 12-19)

The roles and responsibilities are outside the scope of DO-178B.

Quality Assurance Planning (MOD 20)

Quality Assurance is discussed in RTCA 8. Although DO-178B does not require the quality system employed to be compliant to ISO 9001, a planned software quality assurance process must exist.

Documentation (MOD 21)

Documentation is not required to conform to a specific standard, although a defined set of documents and contents is given (RTCA 11). Documentation required by DO-178B corresponds to documentation required by Def Stan 00-55 except in the following cases, where DO-178B provides no equivalent:

- i) Software Safety Plan;
- ii) Software Safety Case (the Software Accomplishment Summary, RTCA 11.20, serves a similar purpose, although the content differs substantially);
- iii) Software Safety Records Log;
- iv) Software Safety Audit Plan;

- v) Software Safety Audit Report;
- vi) Software Risk Management Plan;
- vii) Validation aspects of Software Verification and Validation Plan;
- viii) Acceptance Test Specification;
- ix) Acceptance Test Report;
- x) User Manuals; and
- xi) Software Maintenance Plan.

Development Planning (MOD 22)

RTCA 4 and RTCA 11.2 cover development planning. DO-178B does not require the inclusion of metrics in the software development plan for the purpose of providing evidence of the suitability of the process.

Project Risk Planning (MOD 23)

Project risk planning is not required by DO-178B.

Verification and Validation Planning (MOD 24)

Verification Planning is considered in RTCA 4 and RTCA 11.3. Acceptance criteria for SRS items are not defined in the DO-178B verification plan. Instead, this information is defined within the Software Verification Cases and Procedures document.

Configuration Management (MOD 25)

Configuration management is considered in RTCA 7 and RTCA 11.4.

DO-178B meets all of Def Stan 00-55's requirements for configuration management, except for the requirement that the configuration management system conforms to Def Stan 05-57, and that the configuration management system is automated.

Selection of Methods (MOD 26)

DO-178B does not require formal methods for software specification and design (RTCA 12.3.1 discusses the use of formal methods as a means to gain extra certification credit). The requirements for static analysis are much stronger in Def Stan 00-55 (the use of static analysis is implied by the requirements of RTCA 6.3).

Code of Design Practice (MOD 27)

The code of design practice corresponds to the software development standards defined in RTCA 11.7.

Selection of Language (MOD 28)

Language and tool selection is considered in RTCA 4.4.2. DO-178B does not include the following requirements with respect to selection of programming language:

- i) that the language is strongly typed, block structured, has a formally defined syntax and has predictable program execution;

- ii) that the choice of programming language be justified; and
- iii) that high level languages are preferred to low-level languages.

The following requirements do not exist in DO-178B with respect to compilation system:

- i) that a tool validation certificate is required (although DO-178B does require that tools be qualified); and
- ii) that safety assurance be performed upon the compilation system.

Selection of Tools (MOD 29)

Tool selection is considered in RTCA 4 and RTCA 12.2. Although DO-178B does not require the demonstration that no tool can threaten the safety integrity of the SRS, it is required that every tool used is qualified (depending upon whether it is a development tool or a verification tool).

A number of considerations for tool selection are included in DO-178B. However, the following are not included, but are required by DEF STAN 00-55:

- i) usability;
- ii) interoperability;
- iii) stability
- iv) commercial availability;
- v) maintenance support; and
- vi) familiarity to the design team.

Use of Previously Developed Software (MOD 30)

Previously developed software is considered in RTCA 12.1. The following requirements are not included:

- i) requirements considering the presence of unreachable code in delivered equipment; and
- ii) requirements for previously developed software not developed to the requirements of DO-178B.

Under Def Stan 00-55, in-service history may be used for software not developed to the standard. However, under DO-178B, the only permitted use for in-service history is in gaining certification credit. These may in practice achieve the same result.

Use of Diverse Software (MOD 31)

Multiple-version dissimilar software is considered in RTCA 2.3.2 and RTCA 12.3.3. Under DO-178B, multiple version dissimilar software requires that at least one of the parallel components has the DAL associated with the most severe failure category. Under Def Stan 00-55, all diverse components may be at a lower level of software integrity provided that the combination of components provides the appropriate integrity for the system as a whole.

Development Principles (MOD 32)

RTCA 4 and RTCA 5 cover development issues.

DO-178B does not require the scheduling of the software lifecycle to be defined.

For the software requirements and software design, DO-178B does not require the following:

- i) safety functions and safety properties to be explicitly identified;
- ii) a formal representation of the software specification and the software design;
- iii) a formal specification and formal design of any data on which the SRS relies;
- iv) explicit recording of assumptions; and
- v) explicit recording of reasoning behind decisions.

Verification activities relating to formal specifications and designs are not performed under DO-178B (these include syntax and type checks, consistency checks with informal parts and proof obligations).

Under DO-178B, for the correction of errors, it is not explicitly necessary to consider:

- i) the effect of the anomaly if not corrected;
- ii) the adequacy of the verification process; and
- iii) the additional verification required after correction of the anomaly.

Software Requirement (MOD 33)

RTCA 5.1 covers the software requirements process.

DO-178B does not require:

- i) that verification be performed on the requirements to determine that there is sufficient information to produce a software safety case; and
- ii) that the assignment of the software level be verified.

Specification Process (MOD 34)

RTCA 5.1 covers the software specification process.

DO-178B does not require executable prototypes and formal verification for the specification. Validation of the specification is outside the scope of DO-178B.

Design Process (MOD 35)

RTCA 5.2 covers the software design process.

RTCA 5.2 does not require that:

- i) the design provides justification to show that its specification can be met in terms of performance;
- ii) the design be shown to be consistent with the non-functional requirements; and

- iii) performance analysis of the software design be conducted.

Coding Process (MOD 36)

The coding process corresponds to RTCA 5.3. In this section, the following are not required:

- i) performing static analyses and formal verification;
- ii) justification of object code correctness through formal mappings, static analysis or proof obligations;
- iii) providing evidence of low error rates of the compiler; and
- iv) scrutinising the source for potentially dangerous syntax.

Testing and Integration (MOD 37)

RTCA 6 covers issues of testing and integration.

DO-178B does not explicitly require that testing demonstrates dynamic and performance requirements. The scope of the testing is not required to be defined or justified. Metrics for testing are defined by the standard, rather than on a project basis. Justification for metrics and criteria are not required.

In unit testing, there is a large overlap between testing requirements as defined by the following table:

Test type	Def Stan 00-55	DO178B
Structural coverage	All source code statements and branches	has degree of coverage appropriate to software level.
Equivalence classes and boundary variables	All source code variables	Real and integer Input variables
Equivalence class of invalid values	No requirement	Real and integer variables
Loops, number of iterations	All source code loops, 0, 1 intermediate number and maximum number of times	time related functions, multiple iterations, attempt to compute out-of-range loop count values
Boolean variables	All source code Boolean with true and false values	No requirement
Predicates	All feasible combinations	verify variable usage and Boolean operators (MC/DC at Level A)
Enumerated types	all source code variables set to each possible value	No requirement
Special cases	e.g. variables and expressions that can take the value zero	No requirement

Test type	Def Stan 00-55	DO178B
System initialisation	No requirement	During abnormal conditions
Incoming data		Consider possible failure modes
Non-numerical outputs	All tested	No requirement
State transitions	No requirement	Exercise all normal transitions, attempt not permitted transitions
Non functional requirements	All tested	Arithmetic overflow and exceeded frame times mechanisms work correctly, execution time requirements.
Stress tests	performed	Not required

Table 7.1. Comparison of Testing Requirements

Validation tests are outside the scope of DO-178B.

Certification (MOD 38)

Certification issues are discussed in MOD 9 and MOD 10. However, the certification or approval process (and the requirements imposed by the authorities) is likely to be significantly different.

Acceptance (MOD 39)

Acceptance is outside the scope of DO-178B.

Replication (MOD 40)

Replication is considered as part of the configuration management process, and is discussed under RTCA 7. DO-178B permits a detection mechanism for detecting binary image corruption, which must have the same software level as the most severe software level contained within the binary image. Def Stan 00-55 explicitly requires that binary image comparison is performed against the master copy of the image (held under configuration control).

DO-178B does not require a safety analysis of the replication process.

User Instruction (MOD 41)

User instructions are not considered in DO-178B.

In-service Operation (MOD 42)

In-service operation issues are not considered in DO-178B.

A.3.3 Comparison of Def Stan 00-55 against DO-178B

This section describes activities and data that are required by DO-178B, but are either not present in or not required by Def Stan 00-55. Some of the activities described here, particularly those pertaining to safety management, are covered by Def Stan 00-56.

SYSTEM ASPECTS RELATING TO SOFTWARE DEVELOPMENT (RTCA 2)

Information Flow between System and Software Life Cycle Processes (RTCA 2.1)

Issues of information flow between system and software lifecycles are outside the scope of Def Stan 00-55.

Failure Condition and Software Level (RTCA 2.2)

Under DO-178B, multiple version dissimilar software requires that at least one of the parallel components has the software level associated with the most severe failure category. Under Def Stan 00-55, all diverse components may be at a lower level of software integrity.

Def Stan 00-55 does not require that development standards differ in diverse implementations.

System Architectural Considerations (RTCA 2.3)

Specific issues of architecture are outside the scope of Def Stan 00-55, with the exception of multiple-version dissimilar software (diverse software), which is considered by MOD 31.

SOFTWARE LIFECYCLE (RTCA 3)

MOD 32.1 covers all software lifecycle requirements defined by DO-178B.

SOFTWARE PLANNING PROCESS (RTCA 4)

MOD 4 describes software planning. Sections 9.3.1 to 9.3.5 below discuss planning in greater detail.

Software Planning Process Activities (RTCA 4.2)

Def Stan 00-55 does not permit deactivated code and user-modifiable code.

Software Plans (RTCA 4.3)

The plan for software aspects of certification is not required by Def Stan 00-55. All other requirements of this section are met.

Software Life Cycle Environment Planning (RTCA 4.4)

Def Stan 00-55 does not require that the effects of optional features of tools, if used, be examined and specified.

The planning process for Def Stan 00-55 is not required to deal with the issues of detecting object code that cannot be directly traced to source code. However, this is dealt with in verification.

Software Development Standards (RTCA 4.5)

Software development standards are discussed in MOD 27. All requirements are met.

Review and Assurance of the Software Planning Process (RTCA 4.6)

Def Stan 00-55 meets all the requirements of RTCA 4.6 for review and assurance.

SOFTWARE DEVELOPMENT PROCESS (RTCA 5)

Software Requirements Process (RTCA 5.1)

MOD 33 considers software requirements. There is no requirement to analyse system interface requirements.

Software Design Process (RTCA 5.2)

The software design process corresponds to MOD 35. Under Def Stan 00-55, the following are not required:

- i) the analysis of derived requirements to ensure higher level requirements are not compromised;
- ii) the monitoring of control flow and data flow; and
- iii) that responses to failure conditions be consistent with the safety related requirements (although it is required that the design be consistent with the fault tolerance requirements).

Def Stan 00-55 does not permit user-modifiable software; therefore guidance on this subject is out of scope.

Software Coding Process (RTCA 5.3)

The software coding process corresponds to MOD 36, which meets all of the requirements.

Integration Process (RTCA 5.4)

Def Stan 00-55 does not cover integration in the sense used in DO-178B (actual merging and loading of the software, as opposed to testing integrated software).

Traceability (RTCA 5.5)

Traceability is considered in MOD 32.3, which meets the requirements of this RTCA 5.5.

SOFTWARE VERIFICATION PROCESS (RTCA 6)

Software Reviews and Analyses (RTCA 6.3)

Sections of Def Stan 00-55 relevant to software reviews and analysis are: MOD 34.6, MOD 35.6, and MOD 36.5.

The following requirements are not present in Def Stan 00-55:

- i) considering the compatibility of the requirements (high or low level) with the target computer;
- ii) considering the accuracy and behaviour of algorithms contained within the requirements (high or low level);
- iii) considering compatibility of the software architecture with the target computer;
- iv) considering partitioning integrity of the software architecture; and
- v) review of the linking and loading data and memory map.

Software Testing Process (RTCA 6.4)

The software testing process corresponds to MOD 37.

DO-178B suggests that using an emulator for software development is beneficial, as some errors are only detected in this environment. However, Def Stan 00-55 discourages this by requiring justification for the correctness of the emulator.

For the comparison of test case selection see section 7.20 of this report.

Under Def Stan 00-55, the following categories of errors are not explicitly considered:

- i) incorrect interrupt handling;
- ii) execution time requirements incorrectly met;
- iii) incorrect response to hardware transients/faults;
- iv) data base contention problems;
- v) inability to detect faults with built-in test;
- vi) errors in hardware/software interfaces;
- vii) incorrect behaviour of feedback loops; and
- viii) stack overflow.

The impact of dead code and deactivated code on the testing is not defined (since there is a requirement not to have any such code).

SOFTWARE CONFIGURATION MANAGEMENT PROCESS (RTCA 7)

Configuration management is considered in MOD 25.

The following requirements do not appear in MOD 25:

- i) that each valid combination of configuration items is treated as a separate configuration item;
- ii) that software product can be identified on a delivered system through physical examination or through software access;
- iii) that all configuration items be traceable to the configuration items from which they are derived;
- iv) that all configuration items are traceable to the output they identify or the process with which they are associated;
- v) that the integrity of the stored data is monitored (specifically addressing concerns of deterioration or destruction of the storage medium); and
- vi) that different data control categories be defined and used.

DO-178B requires that a configuration item shall be configuration identified before it is used or referenced. Def Stan 00-55 only requires that a configuration item be uniquely identified from its time of creation onwards.

SOFTWARE QUALITY ASSURANCE PROCESS (RTCA 8)

All requirements for the software quality assurance process are met by MOD 20.

CERTIFICATION LIAISON PROCESS (RTCA 9)

The use of a Plan for Software Aspects of Certification is not required. Different requirements for data exist between the two standards.

OVERVIEW OF AIRCRAFT AND ENGINE CERTIFICATION (RTCA 10)

Certification is not relevant to Def Stan 00-55.

SOFTWARE LIFE CYCLE DATA (RTCA 11)

Plan for Software Aspects of Certification (RTCA 11.1)

Much of the required information within this item can be derived from 00-55 outputs, with the possible exclusion of the certification considerations. Sources include the Software Safety Case, the Software Safety Plan, the Software Development Plan and the Software Quality Plan.

Software Development Plan (RTCA 11.2)

The contents of the software development plan correspond to the software development plan of Def Stan 00-55.

Software Verification Plan (RTCA 11.3)

This Software Verification Plan corresponds to the contained within Def Stan 00-55 Verification and Validation plan, with following exclusions:

- i) partitioning considerations;
- ii) compiler assumptions; and

- iii) multiple version dissimilar software.

Software Configuration Management Plan (RTCA 11.4)

The Software Configuration Management Plan is equivalent to Def Stan 00-55 Software Configuration Management plan

Software Quality Assurance Plan (RTCA 11.5)

The Software Quality Assurance Plan is equivalent to Software Quality Plan.

Software Requirements Standards (RTCA 11.6)

The Software Requirements Standards are contained within the Code of Design Practice of Def Stan 00-55.

Software Design Standards (RTCA 11.7)

The Software Design Standards are contained within the Code of Design Practice.

Software Code Standards (RTCA 11.8)

The Software Code Standards are contained within the Code of Design Practice.

Software Requirements Data (RTCA 11.9)

Def Stan 00-55 uses the term Software Specification in place of Software Requirements.

Def Stan 00-55 does not (explicitly) cover the following aspects of software requirements:

- i) description of the allocation of system requirements to software (with attention to safety-related requirements and potential failure conditions);
- ii) functional and operational requirements under each mode of operation;
- iii) performance criteria;
- iv) timing requirements and constraints;
- v) memory size constraints;
- vi) hardware and software interfaces;
- vii) failure detection and safety monitoring requirements; and
- viii) partitioning requirements allocated to software, how the partitioned software components interact with each other and the software level(s) of the partition.

Design Description (RTCA 11.10)

The Design Description is equivalent to Software Design in Def Stan 00-55.

Source Code (RTCA 11.11)

Source Code is equivalent to Code in Def Stan 00-55.

Executable Object Code (RTCA 11.12)

There is no equivalent requirement on executable object code in Def Stan 00-55.

Software Verification Cases and Procedures (RTCA 11.13)

Software Verification Cases and Procedures corresponds to the Software Verification and Validation plan, and test record.

Software Verification Results (RTCA 11.14)

Software Verification Results corresponds to the Specification Record, Design Record, Test Record.

Software Life Cycle Environment Configuration Index (RTCA 11.15)

The Software Life Cycle Environment Configuration Index corresponds to the Software Configuration Record.

Software Configuration Index (RTCA 11.16)

The Software Configuration Index corresponds to the Software Configuration Record.

Problem Reports (RTCA 11.17)

The existence of problem reports is implied by Def Stan 00-55, but not required as formal documentation.

Software Configuration Management Records (RTCA 11.18)

Software Configuration Management Records corresponds to the Software Configuration Record.

Software Quality Assurance Records (RTCA 11.19)

Software Quality Assurance Records are not explicitly required.

Software Accomplishment Summary (RTCA 11.20)

The Software Accomplishment Summary is replaced by a stronger requirement for a Software Safety Case.

Additional consideration (RTCA 12)

Use of Previously Developed Software (RTCA 12.1)

There is no requirement in Def Stan 00-55 that previously developed software have traceability from product and lifecycle data of the previous application to the new application.

Tool Qualification (RTCA 12.2)

Classification of software tools into development or verification tools for the purposes of tool verification is not required, instead 00-55 distinguishes between compilation systems and tools used in development.

Software development processes for tools are not required to satisfy the same objectives of development processes for safety related software.

Alternative Methods (RTCA 12.3)

This section of DO-178B is addressed in sections 9.11.4 to 9.11.8 below.

Formal Methods (RTCA 12.3.1)

Formal methods are required by Def Stan 00-55.

Exhaustive Input Testing (RTCA 12.3.2)

Exhaustive Input Testing is not covered by Def Stan 00-55.

Considerations for Multiple Version Dissimilar Software Verification (RTCA 12.3.3)

The requirements for multiple version dissimilar software is not covered by Def Stan 00-55.

Software Reliability Models (RTCA 12.3.4)

Software reliability models are not addressed by Def Stan 00-55.

Product Service History (RTCA 12.3.5)

Product service history is used by the certification authority to determine whether the proposed life cycle is commensurate with the level of software being developed.

In Def Stan 00-55 the Software safety plan shows safety planning and control measures to be employed. The Software safety case presents justification that the software does or will satisfy the safety aspects of the software requirement.

A.4 DERA (1999) - A comparison of Avionics Standards

The objective of this report (Ref. 24) was to examine two points of view:

1. where a system has been developed to UK standards, to what extent does this satisfy the requirements of international standards?
2. where a system is claimed to satisfy international standards, to what extent does this meet the requirements of UK standards?

For the current study programme, item 2 is of most interest.

The paper concluded that “the requirements of the Defence Standards and the other standards are very similar. They have very similar objectives and only differ in the methods they recommend to achieve those objectives. It ought to be the case that full compliance with one set of standards should be acceptable to another certification authority using the other set.” A number of caveats are added which are significant.

In addition the paper states “Where a system has been developed under the international standards, for UK certification additional evidence (such as certification plans and the agreement between the manufacturer and the certification authority) needs to be examined, to ensure that an acceptable development process has actually been followed, as all clauses of the international standards are only recommendations.”

This research only considered the theoretical differences between the standards, and did not address the way they had been applied in practice. Furthermore, the requirements for the re-certification of equipment developed to international standards for UK military applications was not considered. In theory, the additional evidence required should be based on an assessment of what is missing, as opposed to a blanket requirement to carry out additional extensive analysis (such as static analysis).

A.5 Adelard (2001) – COTS and Safety Related Applications

In 2001, Adelard published two reports based on HSE commissioned research into how pre-existing software components may be safely used in safety related programmable electronic systems, in a way that complied with IEC 61508.

2001 - Methods for assessing the safety integrity of safety-related software of uncertain pedigree (SOUP) – Adelard

The first report (Ref. 25) summarises the evidence that is likely to be available in practice to assist in assessing the safety integrity of a safety function. Obviously the focus of the report is on achieving compliance to IEC 61508 and DO-178B is not specifically covered.

The report proposes a safety justification approach for safety related software that is linked to the IEC 61508 safety lifecycle. It concludes “It is noted that there are organisational strategies that can be deployed to mitigate the safety threat from SOUP, particularly by building up an evidence base for the AOUP used in less critical applications, and by sharing information at a sector or national level. Such initiatives should be encouraged”.

2001 – Justifying the use of software of uncertain pedigree (SOUP) in safety related applications - Adelard

The second report (Ref. 26) considered how the available evidence can best be used within the framework on the IEC 61508 safety lifecycle to support an argument for the safety integrity achieved by a safety function. Again, the emphasis is on IEC 61508 and there is no specific mention of DO-178B.

However, the report does review justifying the use of SOUP which has been developed to a safety related standard, and notes that “requirements in standards differ considerably, and typically require considerable interpretation in their requirements, or are so rigorous as to be unachievable in most applications.

A.6 ERA (2002) - Comparison of Def Stan 00-56 and SAE ARP 4761/4754

This report (Ref. 27) focuses on the system level view as addressed by Def Stan 00-56 and the ARPs 4761 and 4754. The objective of the study was to determine if ARP 4754 and ARP 4761 could

provide a model for a guidance document on the application of Def-Stan 00-56 (Issue 2) to military avionic and weapon systems.

The study concluded that there is overlap, but “it was not clear whether the civil aircraft specific ARP requirements are sufficiently robust to ensure compliance with the corresponding Defence Standard 00-56 methodology. Development Assurance Level (DAL) A classification defined in ARP 4754 is not consistent with a SIL 4 system, due to differences in definition, therefore cannot say a SIL 4 is comparable to a DAL A system.” The report continues “A more viable option would be to compare the available safety evidence analysing the development processes and procedures adopted for systems developed to both standards to identify correlation and deficiencies”.

Appendix B

DO-178C

DO-178B was published in 1992 and many advances in technology have occurred since that time. Recognising the need to update the document, RTCA formed Special Committee (SC) 205, Software Considerations in Aeronautical Systems, to update DO-178B to DO-178C and to recommend provisions for accommodating software technology trends. SC-205 is a joint activity with working group WG-71 of the EUROCAE. It is their intention that their combined efforts will help harmonize the development and use of avionics software in the global environment.

SC-205 met for the first time during March 2005 and plans to deliver DO-178C in 2008. The new document is expected to do the following:

- Provide a more consistent application of certification requirements
- Clarify the boundaries between systems and software guidance
- Consolidate numerous regulatory and industry documents
- Reduce certification risks.

However, it is not the intention to completely re-write DO-178B. Indeed, key objectives are to “maintain the current objective-based approach for software assurance” and “maintain the technology independent nature of DO-178B objectives”. These objectives will be met by minimising the changes to the existing text of the guidance document itself. Instead, it is anticipated that the main changes to DO-178B will entail the development of supplementary guidance material to discuss specific technologies, methodologies or approaches.

One of the key criticisms of DO-178B is that it does not embrace current technologies. Obviously, many new technologies have become available since DO-178B was originally published in 1992. For example, there is an over emphasis on dynamic testing as the main approach to verification. Although DO-178B includes objectives for review and analysis to support software testing activities, little guidance on the conduct of these activities is provided in DO-178B. The working groups will considered alternative approaches to verification such as static verification and formal methods.

A web site <http://forum.pr.erau.edu/SCAS> provides access to all discussions and development on the progress being made to develop DO-178C.

Obviously the special committee is in its early days, so it is difficult to fully access the impact of the changes to DO-178B. However, it is clear that the relationship between the system and the software and, it is assumed, the system safety assessment process, is a key focus of the changes and additional guidance in this area will be provided. However, since the committee is also trying to minimise the changes to the existing text, it is not clear how this will be fully addressed.